Journal of the Society of Korea Industrial and Systems Engineering
Vol, 29, No. 2, pp.7－12, June 2006.

병렬기계에서의 스케쥴링에 관한 연구†

김 대 철

한양대학교 경영대학 경영학부

Uniform Parallel Machine Scheduling

Dae-Cheol Kim

School of Business, Hanyang University

This study considers the problem of scheduling jobs on uniform parallel machines with a common due date. The objective is to minimize the total absolute deviation of job completion times about the common due date. This problem is motivated by the fact that a certain phase of printed circuit board manufacturing is bottleneck and the processing speeds of parallel machines in this phase are uniformly different for all jobs. Optimal properties are proved and a simple polynomial time optimal algorithm is developed.

Keywords : bottleneck, uniform parallel machine, scheduling, common due date, PCB

1. 서 론

병렬기계에서의 스케쥴링 문제는 오랫동안 많은 연구의 대상이 되어왔다(Baker, 1974; Cheng and Sin, 1990; Pinedo, 1995). 만약 p_{ij}를 작업 j의 기계 i에서의 가공시간이라고 정의한다면, 서로 다른 세 가지 유형의 병렬기계 스케쥴링 문제는 다음과 같이 나타내어질 수 있다 (Piersma and Van Dijk, 1996):

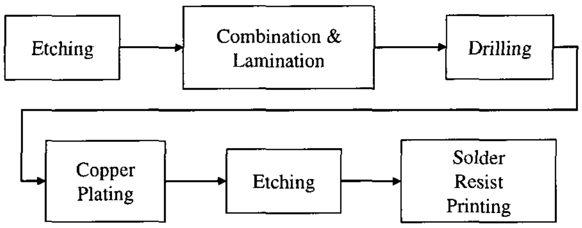
1. 동일 성능(처리속도)의 기계(identical machines) : $p_{ij} = p_{1j}$ for all i and j ;
2. 일정한 비율의 성능차이를 지닌 기계(uniform machines) : $p_{ij} = p_j\ v_i$ for all i and j, v_i는 기계 i의 속도
3. 서로 다른 성능차이를 지닌 기계(unrelated machines) : p_{ij} for all i and j.

본 연구에서는 인쇄회로기판(PCB: printed circuit board) 생산라인의 병목공정(bottleneck operation)에 대한 스케쥴링 문제를 다루고자 한다. PCB 생산라인은 기판에 회로를 각인하는 공정 등 여러 개의 공정으로 구성되어 있다(Yu et al., 2002)(<그림 1> 참조). 그러나, 실제로 제품의 생산성에 영향을 미치는 공정은 단 하나이며 이를 병목공정이라 부른다. PCB 생산라인의 경우, 회로가 각인된 기판에 각종 전자부품칩을 꽂기 위한 구멍을 만드는 드릴링 공정(drilling operation)이 여기에 해당된다. 드릴링 공정은 동시에 여러 개의 드릴링 작업을 수행하기 위하여 여러 개의 병렬 기계들로 구성되어 있는데 이들 병렬기계들의 작업수행성능 즉, 작업처리 속도 등은 서로 다른 것이 보편적이다. 이와 같은 현상은 대개 드릴링 공정에서의 생산능력 확충시 드릴링 기계를 여러 해에 걸쳐 순차적으로 확보하는 가운데서 발생하게 된다. 즉, 새로운 기계 구매시 새로운 성능을 지닌 기계들을 갖추게 되는 경우 이들의 작업처리 속도 등의 수행능력이 서로 다르게 되는 것이다. 이러한 현상에 대한 스케쥴링 문제는 모든 병렬기계들의 성능이 동일한 경우 보다 더 현실을 반영한 모델이지만 문제의 복잡성 때문에 많이 연구되지 못한 것이 사실이다. 본 연구에서는 병렬기계들의 속도가 각 기계마다 서로 일정하게 다른 병

† 이 논문은 2004학년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

렬기계(uniform parallel machines)일 때 이들 공정에서의 스케쥴링 문제를 다루고자 한다.

또한, 최근의 JIT(just-in-time) 개념에 입각하여 각 작업의 납기일자 보다 작업을 미리 완성하는 것은 납기일자 보다 늦게 완성하는 것만큼이나(납기일자까지 재고로 보유해야 되는 것 등 때문에) 바람직하지 못한 것으로 받아들여지는 것이 사실이다. 이를 반영한 스케쥴링 수행능력평가 측정지표의 하나가 MAD(mean absolute deviation) 이며, 이는 스케쥴링의 결과에 의한 작업들의 완료시간이 납기일자로부터 벗어나 있는 정도를 측정한 다(Baker and Scudder, 1990).



<그림 1> 인쇄회로기판 제조공정

Kanet(1981)은 모든 작업이 공통의 납기일자 d를 가지는 경우의 단일기계에서의 MAD를 최소화하는 스케쥴링 문제에 대하여 연구하였다. 모든 작업이 갖는 공통의 납기일자 d가 MS(makespan) 보다 클 경우, 이 문제에 대한 최적의 알고리즘을 개발하였다. Sundararaghavan and Ahmed(1984)는 Kanet(1981)의 연구를 확장하여 단일기계가 아닌 병렬기계들로 구성되어 있는 경우의 스케쥴링 문제에 대하여 최적의 스케쥴이 갖는 특성들을 규명하였다. 또한, 이 특성들을 이용하여 성능이 동일한 병렬기계(identical machines)들에 대한 최적의 알고리즘을 제시하였다. Hall(1986)은 Kanet(1981)이 발견한 스케쥴의 일부 최적조건들을 수정 보완하여, 어떤 한 문제에 대한 여러 해가 존재함을 밝혔고 이 들 해를 구할 수 있는 최적 알고리즘을 제시하였다. Ventura and Kim (2000, 2003)은 성능이 동일한 병렬기계에 대한 기존의 스케쥴링문제를 확장하여각 작업들이 이 들 병렬기계 등의 주자원외에도 보드 또는 AGV등의 제한적인 추가자원을 필요로 할 경우에 대하여 최적알고리즘을 제시하였다.

위에서 언급한 기존 연구에서 살펴본 것과 같이 MAD를 최소화하기 위한 병렬기계의 스케쥴링 문제에 대한 연구는 모두 병렬기계의 성능이 동일하다는 가정을 바탕으로 하고 있다. 하지만 실제 현장에서는 이러한 병렬기계들의 성능이 다른 것이 보다 더 보편적이며 따라서 이 문제에 대한 연구의 필요성이 높아지고 있다.

최근 Palekar et al.(1991), Martello et al.(1997), 그리고 Yu et al.(2002) 등은 서로 다른 능력의 병렬기계에서의 스케쥴링에 관한 연구를 제시하였다. 이 연구들에서는 스케쥴의 우수성을 평가하기 위하여 MS(makespan)과 평균 시스템내에 머문 시간(mean flow time) 등의 규칙적인 지표(regular performance measures)들만 사용하였으며, 본 연구에서 다루고자 하는 MAD 등의 불규칙한 지표(irregular performance measures)들에 대해서는 연구가 이루어지지 않았다(Baker, 1974).

따라서 본 연구에서는 이러한 보다 현실적인 조건을 반영하기 위하여 이들 병렬기계들의 성능이 다르다는 가정 하에서의 MAD(Mean Absolute Deviation)를 최소화하는 스케쥴링 문제에 대하여 연구하고자 한다.

2. Proof of Optimality

전체 작업(jobs)들의 수와 전체 병렬기계들의 수가 각각 n 과 m 일 때, 작업 및 기계들의 집합은 $J = \{J_1, J_2, ..., J_n\}$ 및 $M = \{M_1, M_2, ..., M_m\}$으로 주어졌다고 가정하자. 병렬기계들의 속도가 각 기계마다 서로 일정하게 다르므로 기계 i에서 가공되는 작업 j의 가공처리시간 p_{ij}는 $p_j v_i$로 나타내어 질 수 있으며, 이 때 v_i는 기계 i의 속도이다. 제한이 없는 공통 납기일자(unrestricted common due date)를 d라고 할 때, 일반 보편성의 위배됨이 없이 모든 p_j, v_i와 d의 값들이 정수라고 가정할 수 있다.

또한, 주어진 어떤 하나의 스케쥴에서 s_{ij}를 기계 i에서의 작업 j의 시작시간이라고 하고, c_{ij}를 기계 i에서의 작업 j의 완료시간(즉, $c_{ij} = s_{ij} + p_{ij}$)이라고 정의한다면 MAD는 다음과 같이 나타낼 수 있다.

$$\text{MAD} = \sum_{i=1}^{m}\sum_{j=1}^{n}[max\ \{0,\ c_{ij} - d\} + max\ \{0,\ d - c_{ij}\}]/n$$

본 문제의 최적 스케쥴들이 가지고 있는 2가지 특성들에 대한 증명을 제시하고자 한다. 여기에 제시된 특성들을 밝히기 위하여 사용된 기호 $[i, j]$는 기계 i에서의 스케쥴에서 순서가 j번째인 작업을 나타낸다.

특성 1. 이 문제에 대한 최적스케쥴 중 연속된 2개의 작업들 사이에 유휴시간이 존재하는 최적스케쥴은 존재하지 않는다.

증명: 한 최적스케쥴 σ^*의 기계 i에서의 작업 k와 k+1 사이에 유휴시간이 존재한다고 가정하자. $n(i)$를 스케쥴 σ^*의 기계 i에 계획된 총 작업수라고 하면 k는 1부터 $n(i)$-1 까지의 값을 갖는다(즉, k = 1, 2, ..., $n(i)$-1). 이 때,

$s_{i1} < s_{i2} < \ldots < s_{in(i)}$이라고 가정하면, 작업 k+1의 시작시간은 작업 k 의 완료시간 보다 항상 크다(즉, $s_{ik+1} > c_{ik}$). 이 때, 다음의 2가지 경우를 생각할 수 있다.

i) $c_{ik} < d$일 때,
$\Delta = min\{s_{ik+1},\ d\} - c_{ik}$라고 정의하면, $\Delta > 0$임을 알 수 있다. 여기서, 아래와 같이 정의된 시작시간을 갖는 스케쥴 σ'가 있다고 가정하자 :

$$s'_{ji} = \begin{cases} s_{ij}+\Delta, & \text{for } j=1,2,\ldots,k \\ s_{ij}, & \text{for } j=k+1,\ldots,n(i) \end{cases}$$

$$c'_{ij} = s'_{ij}+p_{ij}, \quad \text{for } j = 1,\ 2,\ \ldots,\ n(_i)$$

$g(r) = max\ \{0,\ d-r\} + max\ \{0, r-d\}$라고 정의하면,
$j = 1, \ldots, k$에 대하여 아래의 식이 성립한다.

$$= max\ \{0,\ d-c'_{ji}\} + max\ \{0,\ c'_{ji}-d\}$$
$$= g(c'_{ji}) < g(c_{ij}).$$

g는 구간 $(-\infty,\ d]$에서는 감소함수이고, $c_{ij} < c'_{ij} \le d$이므로 $g(c'_{ji})$는 $g(c_{ij})$ 보다 작다.

ii) $c_{ik} \ge d$일 경우,
$\Delta = s_{ik+1} - c_{ik}$라고 정의하면, $\Delta > 0$이다. 여기서, 아래와 같이 정의된 시작시간을 갖는 스케쥴 σ'를 고려하자:

$$s'_{ij} = \begin{cases} s_{ij}, & \text{for } j=1,2,\ldots,k \\ s_{ij}-\Delta, & \text{for } j=k+1,\ldots,n_i \end{cases}$$

$$c'_{ij} = s'_{ij}+p_{ij}, \quad \text{for } j = 1,\ 2,\ \ldots,\ n_i$$

이 때, $j = k+1, \ldots, n_i$에 대하여 아래의 식이 성립한다.

$$max\ \{\ 0,\ d-c'_{ji}\} + max\ \{\ 0,\ c'_{ji} - d\ \}$$
$$= g(c'_{ji}) < g(c_{ij}\ ).$$

g는 구간 $[d\ ,\infty)$에서 증가함수이고, $d\ \le c'_{ij} < c_{ij}$이므로, $g(c'_{ji})$ 는 $g(c_{ij})$ 보다 작다.

그러므로, 위의 i)과 ii)의 두 가지 모든 경우에서,

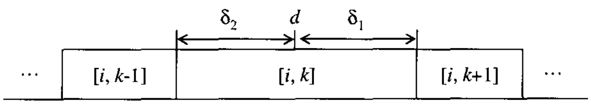
$$\sum_{i=1}^{m}\sum_{j=1}^{n_i}[max\{0, d-c'_{ij}\} + max\{0, c'_{ji}-d\}]$$
$$< \sum_{i=1}^{m}\sum_{j=1}^{n_i}[max\{0, d-c_{ij}\} + max\{0, c_{ij}-d\}].$$

따라서, 스케쥴 σ*를 스케쥴 σ'로 대신할 경우 전체 비용은 감소하고 이 새로운 스케쥴 σ'에는 유휴시간이 없음을 알 수 있다.

특성 2. 이 문제에 대한 최소한 하나의 최적 스케쥴은 납기일자 d에서 정확히 끝내거나 시작하는 작업(job)을 가지고 있다.

증명. 기계 i에서의 임의의 스케쥴 σ_i는 위의 특성 2를 만족하지 않는다고 가정하자. 이 스케쥴 σ_i의 k 번째 작업은 아래 <그림 2>에서와 같이 납기일자 d 보다 전에 시작해서 d 보다 후에 끝난다고 가정하자. 이 때, δ_1과 δ_2는 <그림 2>와 같이 정의한다고 가정하다. 스케쥴 σ_i를 왼쪽으로 이동시켜서 $c_{[i,k]} = d$(또는 $s_{[i,k+1]} = d$)가 되도록 하면 전체 비용에 있어서의 변화 $\Delta z'$는 $\delta_1\{(k\text{-}1) - (n\text{-}k\text{+}1)\}$와 같다. 위와 비슷하게, 스케쥴 σ_i를 오른쪽으로 이동시켜서 $c_{[i,k-1]} = d$(또는 $s_{[i,k]} = d$)가 되도록 하면 전체 비용에 있어서의 변화 $\Delta z''$는 $-\delta_2\{(k\text{-}1) - (n\text{-}k\text{+}1)\}$가 된다. 이 때, 명확한 것은 만약 $(k\text{-}1) \le (n\text{-}k\text{+}1)$이면 $\Delta z' \le 0$이 되고, 만약 $(k\text{-}1) \ge (n\text{-}k\text{+}1)$이면 $\Delta z'' \le 0$이다.

따라서, 특성 2를 만족하지 않는 어떠한 스케쥴이라도 이 스케쥴을 적절히 왼쪽 또는 오른쪽으로 이동시킴으로써 전체 비용측면에서 나쁘지 않은 위의 특성 2를 만족시키는 새로운 스케쥴을 항상 만들어낼 수 있다. 이 것은 납기일자 d에서 정확히 끝내거나 시작하는 작업을 가지는 최소한 하나의 최적 스케쥴이 존재함을 의미한다.



<그림 2> 기계 i에서의 임의의 스케쥴 σ_i

위의 특성들로부터 임의의 스케쥴 σ 대한 목적함수는 다음과 같이 나타낼 수 있다:

$$z(\sigma) = \sum_{i=1}^{m}\Big(\sum_{k=1}^{e_i} C_{E[i,k]} + \sum_{k=1}^{t_i} C_{T[i,k]}\Big) \quad \cdots\cdots (1)$$

이 때, $C_{E[i,k]}$ 는 스케쥴 σ에 있어서 기계 i에 할당된 작업들 중에서 납기일자 d 보다 이전에 시작되기로 계획된 작업들의 순서에서 처음으로부터 k 번째 작업의 비용을 의미하며, $C_{T[i,k]}$는 납기일자 d 보다 이후에 완료되기로 계획된 작업들의 순서에서 끝으로부터 k 번째 작업의 비용을 의미한다. 기계 i 에 할당된 작업들 중에서 e_i는 납기일자 d 보다 이전에 시작되기로 계획된 총 작업수를 그리고 t_i는 납기일자 d 보다 이후에 완료되기로 계획된 총 작업수를 나타낸다. 스케쥴 σ에는 납기일자 d 보다 전에 시작해서 d 보다 후에 끝나는 작업이 없으므로 각 작업들의 가공처리시간 p_{ij}들의 합으로써 나타내어질 수 있다. 보다 자세한 설명은 아래와 같이 표현되어 진다 :

$$
\begin{aligned}
&C_{E[i,1]}:\ v_i\ (\ p_{E[i,2]} + p_{E[i,3]} + \ldots + p_{E[i,k]} + \ldots + p_{E[i,e_i]}\ ) \\
&C_{E[i,2]}:\ \quad v_i\ (\ p_{E[i,3]} + \ldots + p_{E[i,k]} + \ldots + p_{E[i,e_i]}\ ) \\
&\qquad \ldots \qquad \ldots \\
&C_{E[i,k]}:\ \qquad v_i\ (\ p_{E[i,k+1]} + \ldots + p_{E[i,e_i]}\ ) \\
&\qquad \ldots \qquad \ldots \\
&C_{E[i,ei]}:\ \qquad 0.
\end{aligned}
$$

그러므로,

$$
\begin{aligned}
\sum_{k=1}^{e_i} C_{E[i,k]} &= v_i\ (\ 0\ p_{E[i,1]} + 1\ p_{E[i,2]} + 2\ p_{E[i,3]} + \ldots \\
&\quad + (k\text{-}1)\ p_{E[i,k]} + \ldots + (e_i\text{-}1)p_{E[i,e_i]}\ ) \\
&= 0\ p_{E[i,1]} + v_i\ p_{E[i,2]} + 2v_i\ p_{E[i,3]} + \ldots \\
&\quad + (k\text{-}1)v_i\ p_{E[i,k]} + \ldots + (e_i\text{-}1)v_i p_{E[i,e_i]} \qquad \cdots (2) \\
&= w_{i11}\ p_{E[i,1]} + w_{i21}\ p_{E[i,2]} + w_{i31}\ p_{E[i,3]} + \ldots \\
&\quad + w_{ik1}\ p_{E[i,k]} + \ldots + w_{iei1} p_{E[i,e_i]}
\end{aligned}
$$

위와 비슷하게,

$$
\begin{aligned}
&C_{T[i,1]}:\ v_i\ (p_{T[i,1]} + p_{T[i,2]} + \ldots + p_{T[i,k]} + \ldots + p_{T[i,t_i]}\ ) \\
&\qquad \ldots \qquad \ldots \\
&C_{T[i,k]}:\ \qquad v_i\ (\ p_{T[i,k]} + \ldots + p_{T[i,t_i]}\ ) \\
&\qquad \ldots \qquad \ldots \\
&C_{T[i,t_i]}:\ \qquad v_i\ p_{T[i,t_i]}
\end{aligned}
$$

그러므로,

$$
\begin{aligned}
\sum_{k=1}^{t_i} C_{T[i,k]} &= v_i\ (1\ p_{T[i,1]} + 2p_{T[i,2]} + 3p_{T[i,3]} + \ldots \\
&\quad + k\ p_{T[i,k]} + \ldots + t_i\ p_{T[i,t_i]}\ ) \\
&= v_i\ p_{T[i,1]} + 2v_i\ p_{T[i,2]} + \ldots + kv_i\ p_{T[i,k]} + \ldots \\
&\quad + t_i v_i p_{T[i,t_i]} \qquad (3) \\
&= w_{i12}\ p_{T[i,1]} + w_{i22}\ p_{T[i,2]} + \ldots + w_{ik2}\ p_{T[i,k]} + \ldots \\
&\quad + w_{iti2} p_{T[i,t_i]}
\end{aligned}
$$

이 때, 작업 $E_{[i,k]}$는 스케쥴 σ에 있어서 기계 i에 할당된 작업들 중에서 납기일자 d 보다 이전에 시작되기로 계획된 작업들의 순서에서 처음으로부터 k 번째 작업을 의미하며, 작업 $T_{[i,k]}$는 납기일자 d 보다 이후에 완료되기로 계획된 작업들의 순서에서 끝으로부터 k 번째 작업을 나타낸다. 또한, w_{ikl}는 $(k-1)v_i$(만약 $l=1$이면) 또는 kv_i(만약 $l=2$이면)이다.

식 (2)와 식 (3)으로부터 각 기계에서의 위치에 의한 비용들을 감소하지 않는 순서로 나열하면

$$
\begin{aligned}
&M1:\ 0, v_1, v_1, 2v_1, 2v_1,\ \ldots,\ (n-1)/2\quad v_1, \\
&M2:\ 0, v_2, v_2, 2v_2, 2v_2,\ \ldots,\ (n-1)/2\quad v_2, \qquad \cdots\cdots (4) \\
&: \\
&Mi:\ 0, v_i, v_i, 2v_i, 2v_i,\ \ldots,\ (n-1)/2\quad v_i, \\
&: \\
&Mm:\ 0, v_m, v_m, 2v_m, 2v_m,\ \ldots,\ (n-1)/2\quad v_m
\end{aligned}
$$

와 같이 나타내어질 수 있다.

위에서의 x 는 x와 같거나 x 보다 큰 가장 작은 정수를 나타낸다.

Lemma 1. 식 (2)의 비용들을 감소하지 않는 순서로 배열한다. 작업의 가공시간이 가장 큰(LPT : Longest Processing Time)순서 대로 배열한 각 작업들을 앞의 순서대로 정렬된 비용에 맞추어 할당하면 이 스케쥴은 식 (1)을 가장 최소로하는 스케쥴이된다.

증명. 각 작업들이 LPT 순서대로 되어있지 않다고 가정하면, 이 스케쥴에는 각각 기계 a, b의 k, l 번째에 위치하고 $kv_a \le lv_b$와 $p_i < p_j$를 만족하는 한 쌍의 작업(작업 i와 j)들이 존재한다. 이 들 작업들을 서로 맞바꾸어 할당하면 MAD의 변화는 다음과 같다.

$$
\begin{aligned}
&(kv_a\, p_j\ -\ kv_a\, p_i\ +\ lv_b\, p_i\ -\ lv_b\, p_j)/n \\
&= (kv_a\ -\ lv_b)(p_j - p_i)/n \le 0.
\end{aligned}
$$

따라서, LPT 순서로 정렬된 작업들을 증가하는 순서대로 정렬된 위치에 할당하는 것이 최소한 MAD를 증가시키지는 않는 것을 알 수 있다.

3. 알고리즘

위에서 살펴본 Lemma 1 에 의해 본 문제에 대한 최적해가 존재함을 알 수 있다. 또한 여기서 알아둘 것은 특성 1과 2 그리고 Lemma 1을 만족하는 최적 스케쥴은 하나가 아닌 여러 개가 존재할 수 있다는 것이다. 따라서 다음에 제시되는 알고리즘은 n개의 작업과 m개의 기계가 존재할 때 MAD를 최소로 하는 여러 개의 가능한 최적 스케쥴 중에 하나의 최적해를 제시해 준다. 알고리즘에 대한 설명을 위하여 다음의 기호를 정의한다.

w_{ikl}는 기계 i에서 납기일자 전(l=1) 또는 후(l=2)에 종료되는 작업의 k 번째 위치의 비용을 나타내며 이들의 집합 W는 $\{w_{ikl} \mid i \in M,\ k \in J,\ l \in \{1,2\}\}$라고 가정하자. 스케쥴에서 각 작업의 위치를 나타내기 위한 인덱스들의 집합 V를 $\{ikl \mid i \in M,\ k \in J,\ l \in \{1,2\}\}$라고 정의하자. 기계 i에 스케쥴된 작업들의 순서집합을 B_i라고 할 때, 모든 기계(즉, $i=1, 2, \ldots, m$)에서 이 집합의 최종순서에 있는 작업의 완료시간은 납기일 d 와 같다고 할 수 있다. 또한 A_i를 기계 i에 스케쥴된 작업들의 순서집합이라고 가정하면, 모든 기계(즉, $i=1, 2, \ldots, m$)에서 이 집합의 첫 번째 순서에 있는 작업의 시작시

간은 납기일 d와 같다($i = 1, 2, ..., m$). 할당되지 않은 작업들의 집합을 U라고 할 때, 본 연구에서 개발한 알고리즘은 아래와 같이 나타낼 수 있다.

begin
  $B_i \leftarrow \varnothing$, $i = 1, 2, ..., m$
  $A_i \leftarrow \varnothing$, $i = 1, 2, ..., m$
  $U \leftarrow J$
  while ($U \neq \varnothing$) **do**
    remove a job h from U such that $p_h = \max_{j \in U} \{p_j\}$
    remove a position index ikl from V such that $w_{ikl} = \min_{abc \in V} \{w_{abc}\}$
    if $k = 1$ **then**
      insert job h into the last position in B_i
    if $k = 2$ **then**
      insert job h into the first position in A_i
  end
end

수리적 예제

위에서 주어진 알고리즘에 대한 보다 명확한 이해를 위하여, 9개의 작업이 아래 <표 1>과 같은 가공처리시간을 필요로 하고 2대의 기계 ($v_1=1$, $v_2=3$)가 주어졌다고 가정했을 때, 이 문제에 대한 최적해를 나타내고자 한다.

<표 1> 각 작업에 대한 가공처리시간

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
P_j	1	2	4	5	6	7	9	11	13

이 때, 각 기계에서의 위치에 따른 비용을 크기 순으로 나열하면 각각 다음과 같다.

M1 : 0 ($=w_{111}$), 1 ($=w_{121}$), 1 ($=w_{112}$), 2 ($=w_{131}$), 2 ($=w_{122}$), 3 ($=w_{141}$), 3 ($=w_{1321}$), 4 ($=w_{151}$), 4 ($=w_{142}$)
M2 : 0 ($=w_{211}$), 3 ($=w_{221}$), 3 ($=w_{212}$), 6 ($=w_{231}$), 6 ($=w_{222}$), 9 ($=w_{241}$), 9 ($=w_{2321}$), 12 ($=w_{251}$), 12 ($=w_{242}$)

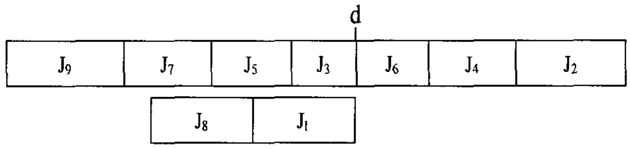
따라서, 위치에 따른 전체 비용들을 감소하지 않는 순서로 나열하면 아래와 같음을 알 수 있다.

0, 0, 1, 1, 2, 2, 3, 3, 3, 3, 4, 4, 6, 6, 9, 9, 12, 12 ······ (5)

또한, 각 작업들을 LPT 순서로 정렬하면 다음과 같다.

$J_9 - J_8 - J_7 - J_6 - J_5 - J_4 - J_3 - J_2 - J_1$ ······ (6)

최종적으로 식 (6)의 LPT 순서대로 작업을 식 (5)의 위치에 차례로 할당하면 MAD를 최소로 하는 하나의 스케쥴을 구할 수 있으며 그 결과는 <그림 3>과 같다.



<그림 3> 알고리즘에 의한 최적 스케쥴링 결과의 예

4. 결 론

본 연구에서 고려된 문제는 기존의 고전적인 병렬기계에서의 스케쥴링 문제들과는 달리 작업들이 납기일자보다 늦게 완료되었을 경우뿐만 아니라 일찍 완료되었을 경우에도 비용이 부과된다. 이 것은 자신의 납기일자 보다 앞서서 생산되는 작업은 납기까지 재고를 발생하므로 JIT 개념에 따라 바람직하지 않다는 현상을 반영한 결과이다. 따라서 기존의 규칙적인 측정지표와는 달리 이러한 비효율적 요인을 반영하여 최소화할 수 있는 MAD를 목적함수로 하고 있다는 것이 최근의 생산현장에서의 경향을 반영한 결과라 할 수 있다.

또한, 각 작업에 요구되는 가공처리 시간은 병렬기계마다 다르다. 이와 같은 현상은 대개 생산능력 확충시 기계를 여러 해에 걸쳐 순차적으로 확보하는 가운데서 발생하게 된다. 즉, 새로운 기계 구매시 새로운 성능을 지닌 기계들을 갖추게 되는 경우 이들의 작업처리 속도 등의 수행능력이 서로 일정하게 다르게 되는 것이다. 이러한 현상에 대한 스케쥴링 문제는 모든 병렬기계들의 성능이 동일한 경우 보다 더 현실을 반영한 모델이지만 문제의 복잡성 때문에 많이 연구되지 못한 것이 사실이다.

본 연구에서는 이 문제에 대한 두 가지 특성 즉, 모든 최적 스케쥴의 각 작업들 사이에는 유휴시간이 존재하지 않는다는 것과, 최소한 하나의 최적 스케쥴은 납기일자 d에서 정확히 끝내거나 시작하는 작업(job)을 가지고 있다는 것을 밝혔고 또한 Lemma 1에서 LPT 순서의 작업들을 비용이 적은 순서의 위치에 차례대로 할당할 경우 최적의 스케쥴을 가질 수 있음을 증명하였다. 이러한 특성들을 이용하여 폴리노미얼시간(polynomial time) 안에 최적해를 구할 수 있는 알고리즘을 개발하였다.

본 연구에서는 병렬기계에서의 스케쥴링에 관한 문제들 중에서 모든 작업들의 납기일자가 동일한 경우의 문제를 다루었다. 납기일자에 관련하여서는 납기일자가 전체 작업들의 가공처리시간의 합보다 작다는 제약을 지닌

경우와 각 작업들의 납기일자가 서로 다른 경우 등의 또 다른 유형의 스케쥴링 문제들이 존재한다. 이러한 경우들에 대한 병렬기계 스케쥴링 문제들도 실제 산업현장에 적용할 수 있는 것들로써 중요한 의미를 지닌다고 할 수 있다. 따라서 이러한 유형의 스케쥴링 문제들에 대한 연구도 흥미있는 앞으로의 연구방향이라고 볼 수 있다.

참고문헌

[1] Baker, K. R., *Introduction to Sequencing and Scheduling*, N.Y., John Wiley, 1974.

[2] Baker, K. R. and Scudder, G. D., "Sequencing with Earliness and Tardiness Penalties: A Review," *Operations Research*, 38 : 22-36, 1990.

[3] Cheng, T. C. E. and Sin, C. C. S., "A state-of-the-art review of parallel-machine scheduling research," *European Journal of Operational Research*, 47 : 271-292, 1990.

[4] Hall, N. G., "Single- and Multi-Processor Models for Minimizing Completion Time Variance," *Naval Res. Logist. Quart.*, 33 : 49-54, 1986.

[5] Kanet, J. J., "Minimizing the Average Deviation of Job Completion Times about a Common Due Date," *Naval Research Logistics Quarterly*, 28 : 643-651, 1981.

[6] Martello, S., Soumis, F., and Toth, P., "Exact and approximation algorithms for makespan minimization on unrelated parallel machines," *Discrete Applied Mathematics*, 75 : 169-188, 1977.

[7] Palekar, U. S., Rama, N., and Toaffe, K., "Duality based relaxations for makespan minimization for unrelated parallel machines," *TIMS/ORSA Bulletin*, 31 (MC2.2) : 21, 1991.

[8] Piersma, N. and Van Dijk, W., "A local search heuristics for unrelated parallel machine scheduling with efficient neighborhood search," *Mathematical Computer Modelling*, 24(9) : 11-19, 1996.

[9] Pinedo, M. and Pinedo, M., *Scheduling: Theory, Algorithm, and Systems*, Englewood Cliffs, N.J., Prentice Hall, 1995.

[10] Sundararaghavan, P. and Ahmed, M., "Minimizing the Sum of Lateness in Single-Machine and Multimachine Scheduling," *Naval Res. Logist. Quart.*, 31 : 325-333, 1984.

[11] Ventura, J. and Kim, D., "Parallel Machine Scheduling about an Unrestricted Due Date and Resource Constraints," *IIE Transactions*, 32(2) : 147-153, 2000.

[12] Ventura, J. and Kim, D., "Parallel machine scheduling with earliness-tardiness penalties and additional resource constraints," *Computers & OR*, 30 : 1945-1958, 2003.

[13] Yu, L., Smith, H. M., Pfund, M., Carlyle, W. M., and Fowler, J. W., "Scheduling of unrelated parallel machines: an application to PWB manufacturing," *IIE Transactions*, 34 : 921-931, 2002.