Journal of the Society of Korea Industrial and Systems Engineering
Vol, 27, No. 4, pp.141－146, December 2004.

병렬기계로 구성된 인쇄회로기판 제조공정에서의 스케쥴링에 관한 연구

김 대 철

충북대학교 경영대학 경영학부 조교수

Unrelated Parallel Machine Scheduling for PCB Manufacturing

Dae-Cheol Kim, Ph.D

Assistant Professor School of Business Chungbuk National University

This research considers the problem of scheduling jobs on unrelated 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 and other production systems is bottleneck and the processing speeds of parallel machines in this phase are different for each job. A zero-one integer programming formulation is presented and two dominance properties are proved. By these dominance properties, it is shown that the problem is reduced to asymmetric assignment problem and is solvable in polynomial time.

Keywords : unrelated parallel machine, scheduling, common due date, PCB

1. 서 론

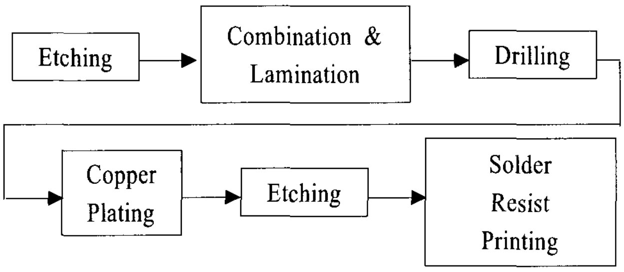
본 연구에서는 인쇄회로기판(PCB : printed circuit board) 생산라인의 병목공정 (bottleneck operation)에 대한 스케쥴링 문제를 다루고자 한다. PCB 생산라인은 기판에 회로를 각인하는 공정 등 여러 개의 공정으로 구성되어 있다 (Yu et al., 2002) (<그림 1> 참조). 그러나, 실제로 제품의 생산성에 영향을 미치는 공정은 단 하나이며 이를 병목공정이라 부른다. PCB 생산라인의 경우, 회로가 각인된 기판에 각종 전자부품칩을 꽂기 위한 구멍을 만드는 드릴링 공정(drilling operation)이 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 다른 병렬기계 (unrelated parallel machines)일때 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 de-

이 논문은 2004년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음.

viation) 이며, 이는 스케쥴링의 결과에 의한 작업들의 완료시간이 납기일자로부터 벗어나 있는 정도를 측정한다 (Baker and Scudder, 1990).

따라서, 본 연구에서는, PCB 생산에 있어서, 각 작업이 동일한 납기 일자를 가지고, 병목공정인 드릴링 공정이 각각의 성능에 차이가 있는 병렬 드릴링 기계들로 구성되어 있을 경우, 이들 병렬기계들에서의 MAD 값을 최소화 하기위한 스케쥴링 문제를 다루고자 한다.



<그림 1> 인쇄회로기판 제조공정

2. 관련 연구

병렬기계에서의 스케쥴링 문제는 오랫동안 많은 연구의 대상이 되어 왔다(Baker, 1974; Chen 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_i 는 기계 i 의 속도
3. 서로 다른 성능차이를 지닌 기계 (unrelated machines) : p_{ij} for all i and j.

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 한문제에 대한 여러 해가 존재함을 밝혔고 이들 해를 구할 수 있는 최적 알고리즘을 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 다른것이 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 병렬기계들의성능이 다르다는 가정 하에서의 MAD를 최소화하는 스케쥴링 문제에 대하여 연구하고자 한다.

3. 수리적 모델 및 특성

3.1 수리적 모델

n개의 작업(jobs)들로 이루어진 집합 $J = \{1, 2, . . . , n\}$이 주어져 있고, 이들이 m개의 병렬기계들의 집합 $M = \{M_1, M_2, . . . , M_m\}$ 에서 가공된다고 할 때, p_{ij} 를 기계 i에서 가공되는 작업 j의 가공처리시간이라 하고 d를 제한이 없는 공통 납기일자 (unrestricted common due date)라고 하자. 이때, 일반 보편성의 위배됨이 없이 모든 p_{ij}와 d의 값들이 정수라고 가정할 수 있다.

또한, 주어진 하나의 스케쥴에서 s_{ij}를 기계 i에서의 작업 j의 시작시간이라고 하고 c_{ij}를 기계 i에서의 작업 j의 완료시간이라 정의하자. 즉, $c_{ij} = s_{ij} + p_{ij}$ 이라고 표현할 수 있다. 우리는 MAD 기준을 사용할 것이며, MAD = $\sum_{i=1}^{m}\sum_{j=1}^{n}[\ max\ \{0,\ c_{ij} - d\} + max\ \{0,\ d - c_{ij}\}\ ]$ 라고 표현될 수 있다.

C_{ijt}를 기계 i에서의 작업 j가 t시점에 가공이 완료되었을 때의 비용이라고 정의한다면, C_{ijt}는 아래와 같이 표현될 수 있다:

$$C_{ijt} = \begin{cases} (d-t) & \text{if } d \geq t, \\ (t-d) & \text{if } d < t, \\ \infty & otherwise. \end{cases}$$

이때, 이 문제에 대한 0-1 정수계획모델은 다음과 같이 나타낼 수 있다:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{t=t_1}^{t_h} C_{ijt} y_{ijt} \quad \cdots\cdots (3.0)$$

s.t.

$$\sum_{i=1}^{m} \sum_{t=t_1}^{t_h} y_{ijt} = 1, \quad j=1, \ldots, n, \quad \cdots\cdots (3.1)$$

$$P_{ij} y_{ijt} \leq \sum_{k=t-p_{ij}+1}^{t} x_{ijk}, \quad i=1, \ldots, m;\ j=1, \ldots, n; \quad t=t_1, \ldots, t_h, \quad \cdots\cdots (3.2)$$

$$\sum_{t=t_1}^{t_h} x_{ijt} = p_{ij}, \quad i=1, \ldots, m;\ j=1, \ldots, n, \quad \cdots\cdots (3.3)$$

$$\sum_{j=1}^{n} x_{ijt} \leqq 1, \quad i=1, \ldots, m;\ t=t_1, \ldots, t_h, \quad \cdots\cdots (3.4)$$

$$x_{ijt} \in \{0, 1\}, \quad i=1, \ldots, m;\ j=1, \ldots, n; \quad t=t_1, \ldots, t_h, \quad \cdots\cdots (3.5)$$

$$y_{ijt} \in \{0, 1\}, \quad i=1, \ldots, m;\ j=1, \ldots, n; \quad t=t_1, \ldots, t_h, \quad \cdots\cdots (3.6)$$

$$y_{ijt} = \begin{cases} 1 & \text{if } job\ j\ is\ completed\ at\ time\ t\ on\ M_i, \\ 0 & otherwise, \end{cases}$$

$$x_{ijt} = \begin{cases} 1 & \text{if } job\ j\ is\ being\ processed\ on\ the\ time\ interval\ (t-1, t)\ on\ M_i, \\ 0 & otherwise. \end{cases}$$

제약식 (3.1)은 각 작업이 어느 한기계에 단한번만 할당되도록 하여준다. 제약식(3.2)와 (3.3)은 각 작업이 자신의 가공시간동안 끊김없이 가공처리가 가능하도록 보장해준다. 끝으로 제약식(3.4)는 각 기계에 있어서의 어떤 한단위 시간에 처리되는 작업은 많아야 하나임을 보장해 준다.

3.2 특 성

본 문제의 최적 스케쥴들이 가지고 있는 2가지 특성들을 밝히고자 한다. 이 특성들을 밝히기 위하여 사용된 기호 $[i,j]$는 기계 i에서의 스케쥴에서 순서가 j번째인 작업을 나타낸다.

특성1. 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 총작업수라고 하면 k는 1부터 n_i-1까지의 값을 갖는다 (즉, k = 1, 2, ..., n_i-1). 이때, $s_{i1} < s_{i2} < \ldots < s_{in_i}$이라고 가정하면, 작업 k+1 의 시작시간은 작업 k의 완료시간 보다 항상 크다 (즉, $s_{ik+1} > c_{ik}$). 이때, 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 알수 있다. 여기서, 아래와 같이 정의된 시작시간을 갖는 스케쥴 σ'가 있다고 가정하자:

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

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

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

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

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

ii) $c_{ik} \geq d$ 일 경우,

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

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

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

이때, j = k+1, ..., n_i에 대하여 아래의 식이 성립한다.

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

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

그러므로, 위의 i) 과 ii)의 두 가지 모든 경우에서,

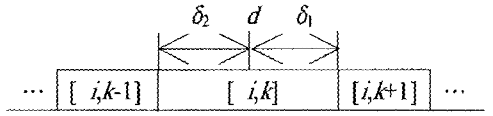
$$\sum_{i=1}^{m}\sum_{j=1}^{n_i}[max\{0,\ d-c'_{ij}\}+max\{0,\ c'_{ij}-d\}]$$
$$<\sum_{i=1}^{m}\sum_{j=1}^{n_i}[max\{0,\ d-c_{ij}\}+max\{0,\ c_{ij}-d\}].$$

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

특성2. 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 특성2를 만족하지 않는다고 가정하자. 이 스케쥴 σ_i 의 k 번째 작업은 아래 <그림2>에서와 같이 납기일자 d보다 전에 시작해서 d보다 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 Δz'는 δ_1 { (k-1) - (n-k+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 Δz" 는 - δ_2{ (k-1) - (n-k+1) } 가 된다. 이 때, 명확한 것은 만약 (k-1)≤(n-k+1) 이면 Δz'≤0 이 되고, 만약 (k-1) ≥ (n-k+1)이면 Δz"≤0 이다.

따라서, 특성 2를 만족하지 않는 어떠한 스케쥴이라도 이 스케쥴을 적절히 왼쪽 또는 오른쪽으로 이동시킴으로써 전체 비용측면에서 나쁘지 않은 위의 특성2를 만족시키는 새로운 스케쥴을 항상 만들어낼 수 있다. 이것은 납기일자 d에서 정확히 끝내거나 시작하는 작업을 가지는 최소한 하나의 최적 스케쥴이 존재함을 의미한다.



<그림 2> 기계 i 에서의 임의의 스케쥴 σi

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

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

이 때, $C_{E[i,k]}$는 스케쥴 σ에 있어서 기계 i에 할당된 작업들 중에서 납기일자 d보다 이전에 시작되기로 계획된 작업들의 순서에서 처음으로부터 k번째 작업의 비용을 의미하며, $C_{T[i,k]}$는 납기일자 d보다 이후에 완료되기로 계획된 작업들의 순서에서 끝으로부터 k번째 작업의 비용을 의미한다. 기계 i에 할당된 작업들 중에서 e_i는 납기일자 d보다 이전에 시작되기로 계획된 총 작업수를 그리고 t_i는 납기일자 d보다 이후에 완료되기로 계획된 총 작업수를 나타낸다. 스케쥴 σ에는 납기일자 d보다 전에 시작해서 d보다 후에 끝나는 작업이 없으므로 각 작업들의 가공처리시간 p_{ij} 들의 합으로써 나타내어질 수 있다. 보다 자세한 설명은 아래와 같이 표현되어 진다 (<그림 3> 참조):

$$\begin{array}{ll}
C_{E[i,1]}: & p_{E[i,2]}+p_{E[i,3]}+\ldots+p_{E[i,k]}+\ldots+p_{E[i,e_i]},\\
C_{E[i,2]}: & \qquad p_{E[i,3]}+\ldots+p_{E[i,k]}+\ldots+p_{E[i,e_i]},\\
 & \qquad \cdots \qquad \cdots\\
C_{E[i,k]}: & \qquad\qquad p_{E[i,k+1]}+\ldots+p_{E[i,e_i]},\\
 & \qquad \cdots \qquad\qquad \cdots\\
C_{E[i,e_i]}: & \qquad\qquad\qquad 0.
\end{array}$$

그러므로, $\sum_{k=1}^{e_i}C_{E[i,k]}$ = 1 $p_{E[i,2]}$ + 2 $p_{E[i,3]}$ + ... + (k-1)

$(k\text{-}1)\ P_{E[i,k]}+\ldots+(e_i\text{-}1)p_{E[i,e_i]}.$ ……………… (2)

위와 비슷하게,

$$\begin{array}{ll}
C_{T[i,1]}: & p_{T[i,1]}+p_{T[i,2]}+\ldots+p_{T[i,k]}+\ldots+p_{T[i,t_i]},\\
 & \qquad \cdots \qquad\qquad \cdots\\
C_{T[i,k]}: & \qquad\qquad p_{T[i,k]}+\ldots+p_{T[i,t_i]},\\
 & \qquad \cdots \qquad\qquad \cdots\\
C_{T[i,t_i]}: & \qquad\qquad\qquad p_{T[i,t_i]}.
\end{array}$$

그러므로, $\sum_{k=1}^{t_i}C_{T[i,k]}$ = 1 $p_{T[i,1]}$ + 2$p_{T[i,2]}$ + 3$p_{T[i,3]}$ +

... + $k\,p_{T[i,k]}$ + ... + $t_i\,p_{T[i,t_i]}$. ……………… (3)

이때, 작업 $E_{[i,k]}$는 스케쥴 σ에 있어서 기계 i에 할당된 작업들 중에서 납기일자 d보다 이전에 시작되기로 계획된 작업들의 순서에서 처음으로부터 k번째 작업을 의미하며, 작업 $T_{[i,k]}$는 납기일자 d보다 이후에 완료되기로 계획된 작업들의 순서에서 끝으로부터 k번째 작업을 나타낸다.

						d				
기계 i	$E_{[i,1]}$	…	$E_{[i,k]}$	…	$E_{[i,e_i]}$	$T_{[i,t_i]}$	..	$T_{[i,k]}$	…	$T_{[i,1]}$

<그림 3> 기계 i 에서의 스케쥴의 예.

4. 할당문제로의 전환

위의 특성 1과 2에 의해서 본 문제는 비대칭 할당문제 (asymmetric assignment problem)로 전환될 수 있다. C_{ijkl}을 기계 i에서의 스케쥴에서 작업 j를 납기일자 d보다 이전에 시작되기로 계획된 작업들의 순서에서 (만약 l = 1 이면) 앞에서 k번째에 할당할 때 발생하는 비용 또는작업 j를 납기일자 d보다 이후에 완료되기로 계획된 작업들의 순서에서 (만약 l = 2 이면) 끝에서 k번째에 할당할 때 발생하는 비용이라고 정의하자. 만약, 작업 j가 기계 i에서의 스케쥴에서 납기일자 d보다 이전에 시작되기로 계획된 작업들의 순서에서 앞에서 k번째 작업이라고 한다면, 등식 (2)에 의해서 이작업에 의해 발생하는 전체비용은 $(k\text{-}1)p_{ij}$가 된다. 이와 비슷하게, 만약 작업 j가 기계 i에서의 스케쥴에서 납기일자 d보다 이후에 완료되기로 계획된 작업들의 순서에서 뒤에서 k번째 작업이라고 한다면, 등식 (3)에 의해서 이 작업에 의해 발생하는 전체비용은 kp_{ij}가 된다. 보다 자세하게 C_{ijkl}는 아래와 같이 나타내어 진다 :

$$C_{ijkl} = \begin{cases} (k-1)p_{ij}, & \text{if } i=1,\ldots,m; j=1,\ldots,n; \\ & \quad k=1,\ldots,n; l=1 \text{ or} \\ kp_{ij}, & \text{if } i=1,\ldots,m; j=1,\ldots,n; \\ & \quad k=1,\ldots,n; l=2 \text{ or} \\ \infty, & \text{otherwise.} \end{cases}$$

이때, 이 문제의 수송네트워크(transportation network) 모델은 n개의 공급지(여기서는 작업들을 나타낸다) 와 mnl 개의 수요지 (i, k, l) (여기서는 기계, 작업의 위치, 그리고 작업이 납기일자 d보다 전에 시작되었는지 또는 뒤에 끝났는지를 나타낸다)들로써 구성된다. 이 수송네트워크모델에서, C_{ijkl}은 단위비용이고 x_{ijkl}은 흐름을 나타내는 의사결정 변수라고 정의하며, 이 두개의 항목들은 모두 아크 (j, (i, k, l))와 관련되어 있다. 따라서, 목적함수 MAD를 최소화하는 스케쥴을 찾고자 하는 본 문제는 다음의 비대칭할당문제의 해를 찾는 것과 동일하게 된다 :

$$Min \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{2} C_{ijkl}x_{ijkl}$$

s.t.

$$\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{l=1}^{2} x_{ijkl} = 1, \; j=1,2,\ldots,n,$$

$$\sum_{j=1}^{n} x_{ijkl} \le 1, \qquad i=1,\ldots,m;\; k=1,\ldots,n;\; l=1,2$$

$$xijkl \;\ge\; 0, \qquad i=1,\ldots,m;\; j=1,\ldots,n;\; k=1,\ldots,n;\; l=1,2,$$

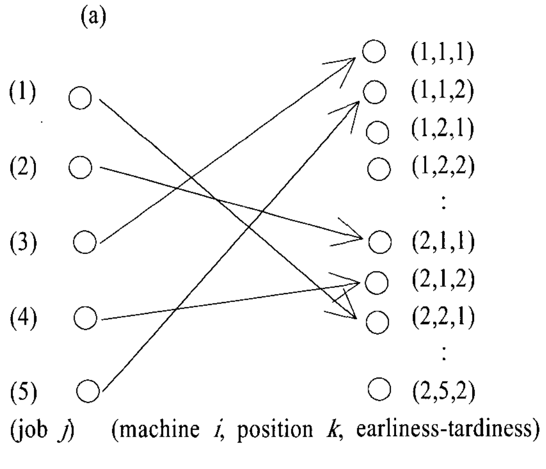
$$x_{ijkl} = \begin{cases} 1 & \text{if } job\ j \text{ is processed on } M_i \text{ in the} \\ & k^{th} \text{ position from the beginning,} \\ & \text{or from the end of the schedule} \\ 0 & otherwise. \end{cases}$$

이것은 기존에 알려진 할당알고리즘 (즉, Bertsekas (1991)의 옥션알고리즘과 Kuhn(1955)의 헝가리안 알고리즘) 들에 의하여 최적의 스케쥴을 구할 수 있음을 의미한다. 보다 상세한 설명을 위하여, 5개의 독립된 작업들과 2개의 병렬기계들이 주어져있을 경우에 대한 문제가 예로써 주어진다. 각 기계에서의 각 작업에 대한 가공처리시간들이 <표 1>에 주어져있다. 각 작업에 동일하게 적용되는 납기일자가 7 이라고 할 때, 이 예제에 대한 최적의 스케쥴은 아래 <그림 4>와 같이 나타내어진다.

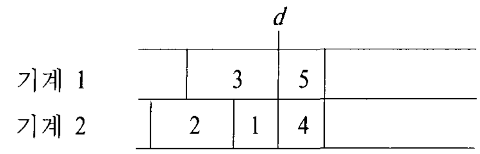
<표 1> 각 작업의 각 기계에서의 가공처리시간.

j i	작업 1	작업 2	작업 3	작업 4	작업 5
기계 1	2	1	3	2	1
기계 2	1	3	2	1	2

(a)



(b)



<그림 4> (a)수송네트워크(최적해에서 $xijkl$ = 1 인 아크들만 표시됨); (b)최적스케쥴

5. 결 론

본 연구에서 고려된 문제는 기존의 고전적인 병렬기계에서의 스케쥴링 문제들과는 달리 작업들이 납기일자보다 늦게 완료되었을 경우 뿐만 아니라 일찍 완료되었을 경우에도 비용이 부과된다. 이것은 자신의 납기일자보다 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 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 정수계획모델을 제시하였고, 모든 최적스케쥴의 각 작업들사이에는 유휴시간이 존재하지 않는다는 것과, 최소한 하나의 최적 스케쥴은 납기일자 d에서 정확히 끝내거나 시작하는 작업(job)을 가지고 있다는 특성들을 밝혔다. 또한, 이 특성들을 이용하면 이러한 복잡한 문제가 폴리노미얼시간(polynomial time)안에 최적해를 구할 수 있는 알고리즘이 존재하는 비대칭할당 문제로 전환할 수 있음을 밝혔다.

본 연구에서는 병렬기계에서의 스케쥴링에 관한 문제들중에서 모든 작업들의 납기일자가 동일한 경우의 문제를 다루었다. 납기일자에 관련하여서는 납기일자가 전체 작업들의 가공처리시간의 합보다 작다는 제약을 지닌 경우와 각 작업들의 납기알자가 서로 다른 경우 등의 또 다른 유형의 스케쥴링문제들이 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**, pp. 22-36, 1990

[3] Bertsekas, D.P., Linear Network Optimization, Algorithms and Codes. MIT Press, Cambridge, Mass. 1991

[4] 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

[5] Kuhn, N.W., "The Hungarian Method for the Assignment Problem." Naval Research Logistics Quarterly, **2**, pp. 83-97, 1955

[6] Martello, S., Soumis, F. and Toth, P., "Exact and approximation algorithms for makespan minimization on unrelated parallel machines," Discrete Applied Mathematics, **75**, pp. 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), pp. 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), pp. 11-19, 1996

[9] 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**, pp. 921- 931, 2002