Journal of the Korean Institute
of Industrial Engineers
Vol. 15, No. 2, Dec., 1989

네트워크 흐름 모형을 이용한 併行機械 시스템의 스케쥴링†

鄭南基* · 朴衡圭* · 梁元燮*

Parallel Machine Scheduling with an Aid of Network Flow Model

Nam-Kee Chung, Hyung-Kyu Park and Won-Sub Yang

Abstract

The problem of scheduling n-jobs on m-uniform parallel machines is considered, in which each job has a release time, a deadline, and a processing requirement. The job processing requirements are allocated to the machines so that the maximum of the load differences between time periods is minimized. Based on Federgruen's maximum flow network model to find a feasible schedule, a polynomially bounded algorithm is developed. An example to show the effectiveness of our algorithm is presented.

1. 緖 論

併行機械 시스템은 동일한 종류의 作業을 처리하는 機械가 다수이며, 각 作業이 임의의 機械에 배정되어 처리될 수 있는 생산 시스템이다. 이것은 공정별 배치의 생산시스템에서 볼수 있으며, 슈퍼 컴퓨터나 분산처리 시스템을 운용할 때에도 유사한 형태를 볼수 있다.

이 논문에서는 機械의 사용시간이나, 作業의 작업요청 시간에 제약이 있는 併行機械 시스템에서 시간대별 작업량이 보다 平準化된 스케쥴 방법을 구하고자 한다. 각 作業은 서로 독립적이며, 한 機械가 동시에 2개 이상이 作業을 처리할 수 없고, 作業의 수행이 어느때든지 중단될 수 있으며, 도중에 중단된 作業은 다른 機械에서도 준비시간 없이 다시 시작될 수 있다고 가정한다.

* 全南大學校 工科大學 産業工學科

† 本 硏究는 1988年度 文敎部의 學術硏究 助成費의 支援에 의해 遂行되었음.

단, 作業의 중단과 재개에 따른 비용은 없다고 가정하자. 즉, m개의 機械(i=1, 2, …, m)가 각각 서로 다른 작업 처리속도 $s_1 \geq s_2 \geq \cdots \geq s_m$을 갖고, n개의 作業(j=1, 2, …, n)은 처리되어야 할 작업량이 p_j이며, 도착하는 시간 r_j와 납기 d_j가 규정되어 있다고 하자. 작업 j가 기계 i에서 처리되는 시간을 t_{ij}라고 하면 $\sum_{i=1}^{m} s_i t_{ij} = p_j$가 성립한다. 이러한 제약조건이 주어져 있을 때, 모든 作業을 각 납기 이전에 처리할 수 있는 가능한 스케쥴의 존재 여부는 Martel[8]이 제시한 Polymatroidal 네트워크 흐름 모형에 의해 판단되어질 수 있다(Polymatroidal 네트워크 흐름 모형은 네트워크에서 개개의 arc에 대한 용량제약 대신 arc의 집합에 용량제약이 있는 최대 흐름 문제이다). 또, Federgruen[3]은 Martel의 모형을 개개의 arc에 용량제약이 있는 일반적인 최대흐름 네트워크 모형으로 변환할 수 있음을 보이고 있다. 그는 이를 이용하여, 가능한 스케쥴이 존재하지 않은 경우에라도, 그 중에서 보다 좋은 성과를 달성하는 스케쥴을 구하기 위한 방법들을 제시하고 있다.

우리는 Martel과 Federgruen이 다루었던 스케쥴링 문제에 대해, 가능한 스케쥴을 구하는 데 그치지 않고, 그 중에서 보다 平準化된 스케쥴을 구하는 방법을 제시하고자 한다. 각 作業의 도착시간과 납기를 빠른 순서부터 배열하면 계획기간이 (2n-1)개의 구간으로 나뉜다(機械의 이용시간에 제약이 있으면 機械別 이용시작 시간과 이용 불가능 시간을 作業의 도착시간과 납기와 같이 배열하여, 구간의 수를 증가시킨다. 그러나, 여기서는 機械 이용시간에는 제약이 없다고 가정한다.). 이때 i번째 구간에서의 機械의 총 능력과 이 구간에서 수행해야 할 작업량을 각각 w_i, q_i, i=1, 2, …, 2n-1이라 하자. 결국 平準化된 스케쥴이란, 총 작업량 $\sum_{j=1}^{n} p_j$가 (2n-1)개의 구간에 공평하게 배분된 것으로, q_i/w_i의 비율이 가능한 일정하게 된 것이라 할 수 있다. 이것은 총 자원 $\sum_{j=1}^{n} p_j$를 q_i, i=1, 2, …, 2n-1로 적절하게 배분하는 일종의 자원할당(resource allocation) 문제이다.

平準化의 尺度로는, 자원 할당문제에서 사용되는 여러가지 尺度중, 여기서는 다음 3가지를 고려한다.

(1) (maximin 기준) q_i/w_i의 최소값을 최대화 한다.

(2) (minimax 기준) q_i/w_i의 최대값을 최소화 한다.

(3) (min(max-min) 기준) q_i/w_i의 최대값과 최소값의 차이를 최소화 한다.

먼저, 제 2절에서는 가능한 스케쥴의 존재여부를 확인하기 위한 Federgruen의 최대 흐름 네트워크 모형을 살펴보며, 제 3절에서는 이 모형을 이용하여 maximin 기준과 minimax 기준에 의한 각각의 스케쥴링 방법을 검토하고, min(max-min) 기준에 의한 스케쥴 방법을 제시한다.

2. 최대 흐름 네트워크 모형

併行機械 시스템에서 납기 이전에 作業이 처리될 수 있는 가능한 스케쥴을 다음 2단계의 절차를 따라 정해진다. 먼저, 각 구간 $T_i = t_{i+1} - t_i$, i=1, 2, …, 2n-1에서 作業 j가 수행되어야 할 작업량 d_j^i를 결정한다. 이 작업량은 Horvath 등[6]이 제시한 조건을 만족하는 것으로, 機械가 처리할 수 있는 작업량의 한계치 보다 작아야 한다. 이 조건을 만족하는 d_j^i에 대해서는 구간 T_i에서 가능한 스케쥴이 존재한다. 두번째 단계에서는, 각 구간에 할당된 작업량 d_j^i를 기준으로 각 구간 안에서의 가능한 스케쥴을 구한다. 이것은 모든 作業이 t_i에 동시에 도착하고, t_{i+1}을 동일한 납기로 갖는 경우의 스케쥴링으로서, Gonzalez-Sahni[5]의 해법을 이용하면 $O(n+$

$m\log n$) 시간에 가능한 스케줄을 구할수 있다.

우리는, 각 구간별 작업량을 平準化 시키는 것이 목표이므로, 첫번째 단계에만 관심을 갖는다. Horvath 등의 조건을 만족시키는 작업량을 결정하는 구체적인 방법으로는 Martel[8]의 polymatroidal 네트워크 모형과 Federgruen [3]의 최대 흐름 네트워크 모형이 있다. 그 중에서도 Federgruen의 모형은 네트워크에서의 최대 흐름량 해법을 적용할 수 있으므로 매우 유용하다. 이 모형은 node N 과 arc E로 구성된 네트워크(N, E)로 표현되며, 그림 1과 같다.

node는 다음과 같은 수준 0에서 수준 3까지의 node 집합으로 구성되며, $N=0\cup N_1\cup N_2\cup N_3$이다.

수준 0: 흐름의 원천으로 0이라 표시된 node 한개로 구성된다.

수준 1: 각 作業에 대응되는 n개의 node N_1으로 구성된다(이것을 "작업 node"라 한다).

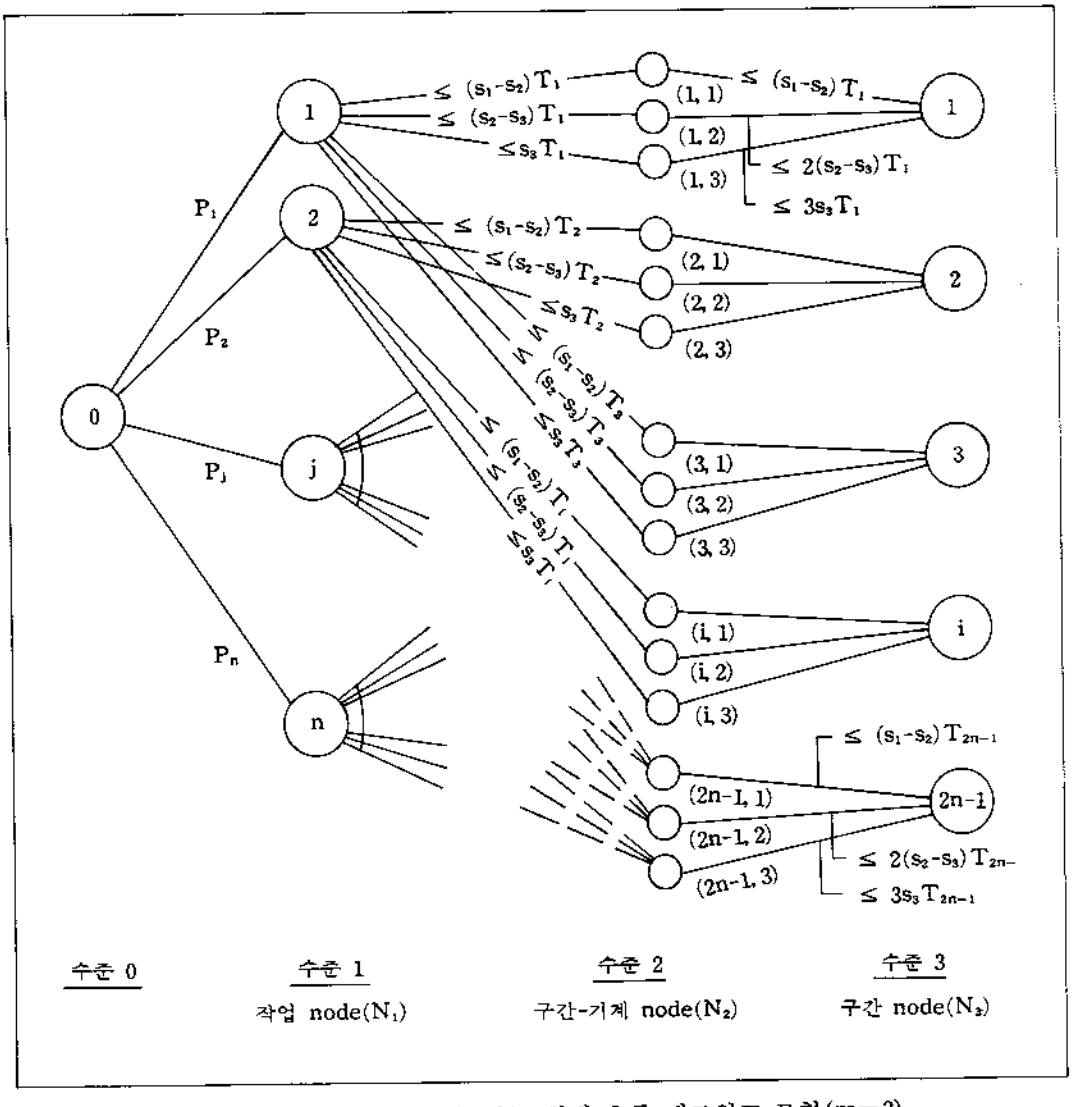

그림 1. 가능한 스케줄을 찾는 최대 흐름 네트워크 모형(m=3)

수준 2: 각 作業 구간별 機械에 대응되는 (2n-1)m 개의 node N_2로 구성된다(이것을 "구간-기계 node"라 하고, i번째 구간의 구간-기계 n-node는 N_{2i}, i=1, 2, …, 2n-1로 표시한다).

수준 3: 각 作業 구간에 대응되는 (2n-1)개의 node N_3로 구성된다(이것을 "구간 node"라 한다).

arc는 다음 (1)-(4)와 같이 각 node 수준간을 연결하며, 흐름량의 한계가 주어진다.

(1) 0으로부터 각 作業 node와 연결된 arc들로써 용량이 각각 p_j이다.

(2) 作業 node로부터 그 作業이 처리 가능한 구간의 구간-기계 node로 연결되며, 機械의 작업 처리속도가 s_r이라 하면, 용량은 $(s_r-s_{r+1})T_i$이다.

(3) 각 구간-기계 node로부터, 그 구간에 대응되는 구간 node로 연결되며, 용량은 $r(s_r-s_{r+1})T_i$이다.

x_{ij}를 임의의 arc$(i, j)\in E$에서의 흐름량이라 하고, 그 arc의 흐름 용량을 u_{ij}라 하자. 그러면, 이 네트워크(N, E)에서의 흐름 제약은,

$$\Sigma_{j\in N_1}x_{ji}-\Sigma_{j\in N_3}x_{ij}=0,\ i\in N_2 \quad \cdots\cdots (2.1)$$

$$\Sigma_{i\in N_2}x_{ji}\le p_j,\ j\in N_1 \quad \cdots\cdots (2.2)$$

$$0\le x_{ij}\le u_{ij},\ (i, j)\in E \quad \cdots\cdots (2.3)$$

이며, 가능한 스케줄이 존재하기 위한 필요충분조건은 네트워크(N, E)에서 최대 흐름량이 $\Sigma_{j\in N_1}p_j$인 것으로 요약된다[3]. 또, 최대 흐름량은 0으로부터 N_3로 연결된 흐름 확대경로(flow augmenting path)를 반복적으로 찾아감으로써 구해지며, $0(mn^3)$의 시간안에 구해진다[3]. 최대 흐름량이 $\Sigma_{j\in N_1}p_j$일 때, 作業 j가 T_i 구간에서 처리되어야 할 작업량 d_j^i는 $\Sigma_{k\in N_{2i}}x_{jk}$, $j\in N_1$이며, 각 구간 T_i에 할당되는 작업량을 q_i라 하면, $q_i=\Sigma_{j\in N_1}d_j^i=\Sigma_{j\in N_{2i}}x_{ji}$이다.

3. 平準化된 스케줄을 위한 解法

가능한 스케줄이 존재하기 위해서는 각 구간의 작업량 $q_i=\Sigma_{j=1}^{n}d_j^i$가 機械의 능력 한계를 벗어나지 않아야 하지만, 작업처리의 효율을 높이기 위해서는 이와 함께 작업량을 기계 능력 한계의 크기에 따라 각 구간에 적절히 배분하여, 작업량의 변동을 줄이는 것이 필요하다. 각 구간의 기계 능력 한계 w_i는 구간의 길이 T_i와 기계의 이용 가능 상황에 따라 달라지며, 어느 구간에서나 모든 機械가 이용 가능하다면, w_i는 T_i에 비례하게 된다. 이러한 문제는, $\Sigma_{j=1}^{n}p_j$를 w_i라는 가중치를 고려하여 각 구간에 공평하게 할당하는 일종의 자원 할당 문제이다. 자원 할당시에 만족시켜야 할 제약조건이 2절에서 본 바와 같은 네트워크 모양으로 나타나므로, 이것은 특별히 네트워크 환경에서의 자원 할당 문제라 할수 있다. 이러한 문제는 Megiddo[9]에 의해 처음 제기되었으며, 平準化의 尺度를 달리하여 서로 다른 해법들이 제시되었다[1, 2, 3, 4, 7].

각 구간별 작업량이 w_i를 고려하여 완전히 平準化된 것으로는 q_i/w_i, i=1, 2, …, 2n-1이 일정한 경우를 생각할 수 있다. 그러나, 실제 문제에서는 완전히 平準化된 가능한 스케줄은 존재하기 어려우므로, 보다 완화된 목표로써 다음 기준들을 생각한다.

(1) [maximin 기준] $\max(\min_{i=1,\cdots,2n-1} q_i/w_i)$ …… (3.1)

(2) [minimax 기준] $\min(\max_{i=1,\cdots,2n-1} q_i/w_i)$ …… (3.2)

(3) [min(max−min) 기준] $\min(\max_{i=1,\cdots,2n-1} q_i/w_i-\min_{i=1,\cdots,2n-1} q_i/w_i)$ …… (3.3)

여기에서 가능한 스케줄을 위한 제약조건을 고려하면, 平準化된 스케줄을 위한 문제는 다음

과 같이 정식화 된다.

목적함수 : (3.1) (또는 (3.2) 나 (3.3))

제약조건 : (2.1), (2.2), (2.3),

$\Sigma_{j\in N_i} x_{ji} = q_i,\ i\in N_3.$

위 문제에 대해 목적함수가 (3.1)인 경우에는 Brown[1]의 해법을 적용할 수 있고, 목적함수가 (3.2)인 경우는 Ichimori 등[7]의 해법을 적용할 수 있다. 우리는 먼저 이 두 해법을 平準化 대상을 축소하는 관점에서 새롭게 해석하고, 이를 바탕으로 (3.3)을 목적함수로 하는 새로운 해법을 제시하고자 한다.

3.1. maximin 기준과 minimax 기준에 대한 해법

이 두 기준에 대한 Brown의 해법과 Ichimori 등의 해법은 q_i/w_i가 모든 구간에서 일정하도록, 즉 완전히 平準化 되도록 작업량을 각 구간에 배정하는 것을 목표로 한다. 이 목표가 달성되지 못할 경우, 平準化 대상 구간을 축소시킨 후, 다시 이 구간에 대해 새로운 목표치로서 완전히 平準化된 작업량을 정한다. 두 해법의 차이는 이렇게 반복되는 과정중에서 대상 구간의 범위를 축소시키는 방법이라 할 수 있다.

네트워크 모형에서 각 구간은 $i\in N_3$에 대응되며, 平準化 대상 구간은 $T\subseteq N_3$이다. 이 두 해법은 $T=N_3$로부터 시작한다. 이때, 平準化 대상 총 작업량은 $F=\Sigma_{j\in T}\Sigma_{i\in N_i} x_{ij}$이다. 각 구간이 완전하게 平準化된 작업량을 q^*_i, $i\in T$라 하면, $\Sigma_{i\in T} q^*_i = F$이고, $q^*_i/w_i = q^*_\alpha/w_\alpha$, $\alpha,\ i\in T$이므로, $q^*_\alpha = w_\alpha F/\Sigma_{i\in T} w_i$이다.

그런데, 이것은 네트워크 모형에서의 제약조건들, 즉 흐름량 보존이나 arc의 용량 등을 무시했을 때 달성 가능한 平準化 목표라 하겠다. 실제로는 이러한 제약들 때문에 $q_i < q^*_i$이거나 $q_i > q^*_i$일 경우가 많으므로, 전자를 과소 공급구간, 후자를 과잉 공급구간이라 하자. 과잉 공급구간에 할당된 과잉 공급량만큼을 제약조건의 범위내에서 과소 공급구간으로 재 분배하면, 보다 平準化된 스케쥴이 가능하다. 이 재분배 과정은 최대 흐름 네트워크 해법을 이용한다. 이상의 설명을 다음과 같이 LEVEL(T)로 정리해 둔다.

LEVEL(T)

(1) $F=\Sigma_{j\in T}\Sigma_{i\in N_2} x_{ij}$와 $q^*_\alpha F/\Sigma_{i\in T} w_i$, $\alpha\in T$를 계산한다.

(2) 최대 흐름 네트워크 해법을 이용하여 과잉 공급량을 과소 공급구간에 재 분배한다.

해법 MAXMIN은 maximin 기준에 따라 平準化된 스케쥴을 구하기 위한 것이며, LEVEL(T)를 활용하여 다음과 같이 간단히 표현된다([1] 참조).

해법 MAXIMIN

단계 1. 0에서 N_3로 최대 흐름량 M을 구한다. $M < E_{j=1}^{n} p_j$이면 가능해가 존재하지 않으므로 끝난다. 그렇지 않으면 $T=N_3$으로 놓는다.

단계 2. LEVEL(T)를 수행한다.

단계 3. 과소 공급구간이 없으면 끝나고, 그렇지 않으면 $T\leftarrow${과소 공급구간}이라 하고, 단계 2로 간다.

해법 MAXMIN은 그림 2로써 보다 쉽게 설명될 수 있다. y축상의 점 B는 $T=N_3$일 때, 완전히 평준화된 양 q^*_i/w_i이고, 편의상 곡선 ACE가 q_i를 나타낸다고 하자(실제로는 q_i가 $i\in N_3$에 대응하므로 점으로 표현된다). 그러면, BC에 해당하는 x축의 구간은 과잉 공급구간, CD는 과소 공급구간에 각각 해당한다. LEVEL(T)의 (2)에 의해 재분배된 결과는 곡선 FCHG로 나타나며, HD는 아직도 과소 공급

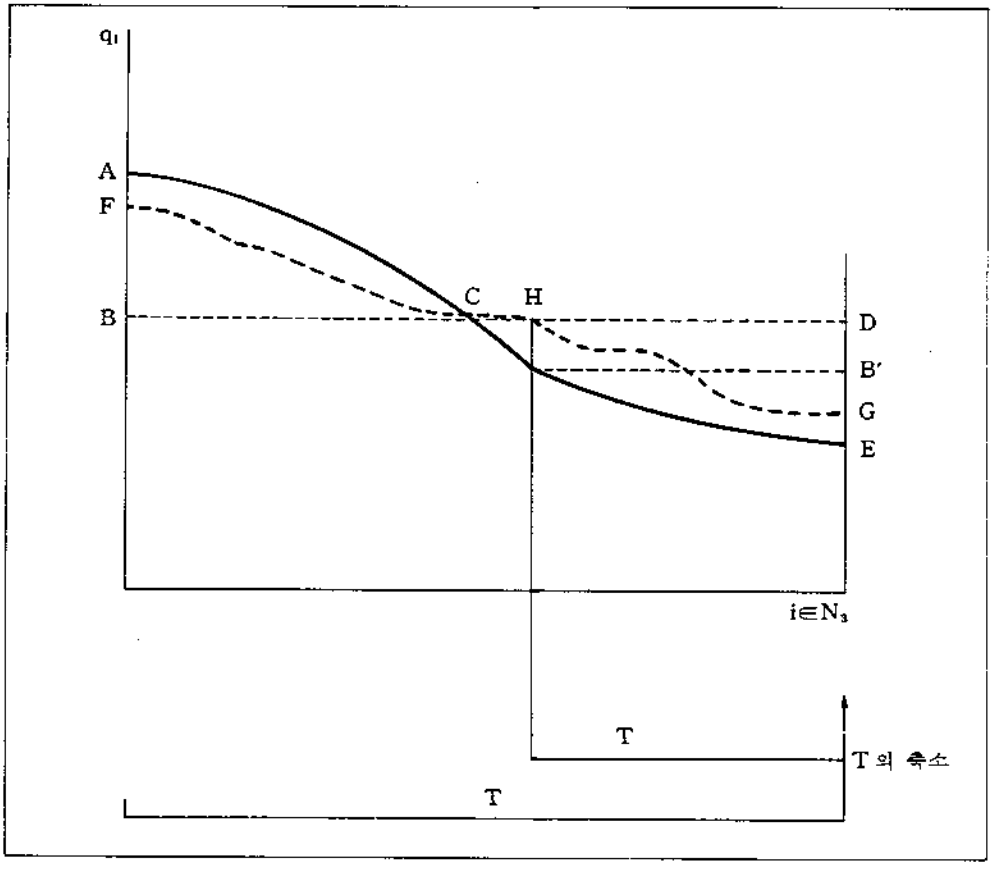

그림 2. 해법 MAXIMIN

구간으로 남는다. 이 구간을 平準化 대상의 새로운 구간으로 정하고, 이때 완전히 平準化된 작업량은 B′가 된다. 따라서 해법 MAXMIN은 과소 공급구간이 없어질 때까지 (i) 과잉 공급량을 과소 공급구간에 재분배 하고 (ii) 과소 공급구간을 새로운 平準化 대상 구간으로 정하는 것으로 요약될 수 있다.

다음으로, minimax 기준에 따라 平準化된 스케줄을 구하기 위한 해법(MINMAX)을 해법 MAXMIN과 비교하여 생각하자. 해법 MINMAX는 과잉 공급구간이 없어질 때까지 (i) 과잉 공급량을 과소 공급구간에 재분배하되 (ii) 과잉 공급구간을 새로운 평준화 대상 구간으로 정한다. 그러므로, 이 해법은 MAXMIN의 단계 3만 수정하여 다음과 같이 표현된다([7] 참조).

해법 MINMAX

단계 1. } MAXMIN과 같음
단계 2. }

단계 3. 과잉 공급구간이 없으면 끝나고, 그렇지 않으면 T←(과잉 공급구간)이라 하고, 단계 2로 간다.

해법 MINMAX를 그림으로 표현하면 그림

3과 같다. B″가 새로운 구간 BH에 대해 완전히 평준화 한 작업량이다.

해법 MAXMIN과 MINMAX의 수행시간은, C(N, E)를 최대 흐름 네트워크 해법의 수행시간이라 할 때, 모두 $0(|N_s| C(N, E))$이다 [1, 7].

3.2. min(max-min) 기준에 대한 해법

平準化의 기준으로서 min(mix-min)은 min(max)-max(min)과 동일하므로, maximin 기준과 minimax 기준을 동시에 고려한 것으로 볼수 있다. 이 기준이 자원할당 문제에서 사용된 것으로는 Zeitlin[10]의 연구가 있으나, 네트워크 환경하에서의 자원할당 문제에 대한 해법은 아직 제시되지 않았다. 우리는 해법 MAXMIN과 MINMAX에 대한 그림 설명을 활용하여 이 기준에 대한 작업량 평준화 해법(MINMAX-MAXMIN)을 보이고자 한다. 그림 2와 그림 3에서 보인 것과 같이, 해법 MAXMIN은 과소공급구간으로 平準化 대상 구간을 축소시켜 가

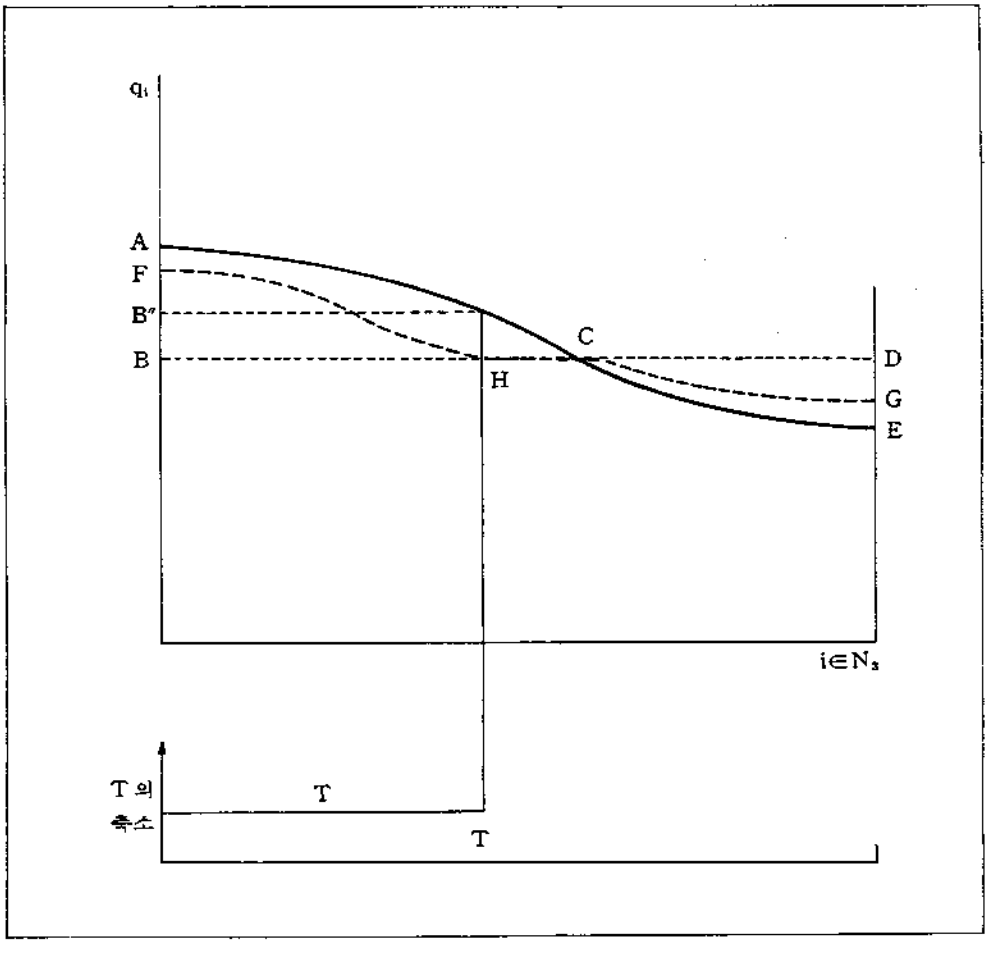

그림 3. 해법 MINMAX

며, 이 구간내에서 작업량을 재분배 하고, 해법 MINMAX는 과잉 공급구간으로 평준화 대상 구간을 축소시켜 작업량을 재분배 한다. 해법 MINMAX-MAXMIN은 기본적으로, 이 두 과정을 함께 수행하는 것이나, 한가지 추가적으로 고려되는 것은 작업량을 재분배 할때 양쪽 구간을 모두 활용하여 과소 공급구간, 또는 과잉 공급구간에 한정되지 않도록 하는 것이다. 이 추가적인 절차가 필요한 것은 다음 그림 4에 의해 확인된다.

기준 (3.3)의 목표는 B″를 초과하는 q_i가 없고, B′ 미만의 q_i가 없을때 B″와 B′의 차를 최소로 줄이는 것이다. abcd를 T_0 구간에서 B″를 목표로 재분배 한 결과라 하고, efgh를 T_1 구간에서 B′를 목표로 재분배한 결과라 하면, 이때 T_{01} 구간에서는 q_i가 재분배 되기 전보다 증가되며(점찍힌 부분), T_{10} 구간에서는 q_i가 재분배 되기 전보다 감소된다(빗금친 부분). 따라서 T_{00} 구간으로부터 T_{10} 구간으로, 또 T_{01} 구간으로부터 T_{11} 구간으로 q_i가 재분배될 가능성이 새로이 생겨난다. 만약, 네트워크의 제약속에서 이것이 이루어진다면 현재의 q_i가 B″ 이

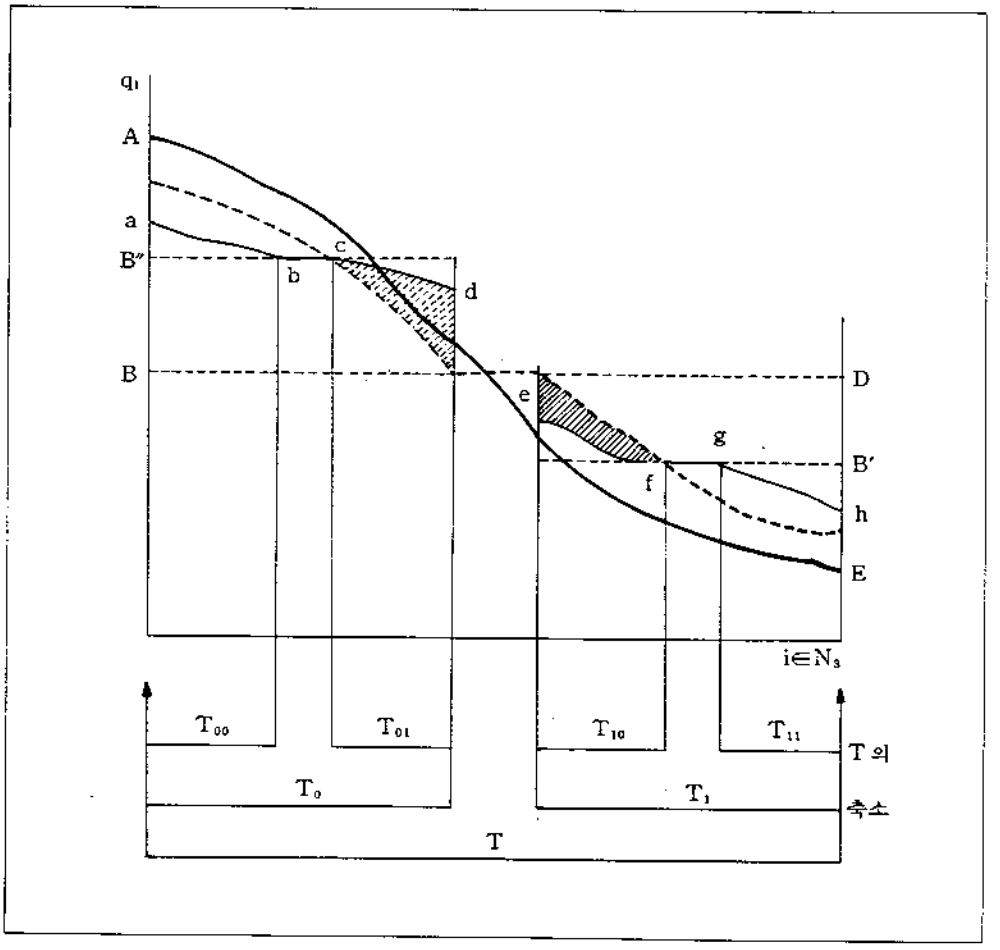

그림 4. 해법 MAXIMIN-MINMAX

하로 혹은 B′ 이상으로 될 수가 있으며, 따라서 B″와 B′의 차는 더 줄어들게 된다. 이와 같은 설명에서 MINMAX-MAXMIN 해법은 개략적으로 다음과 같이 표현된다.

해법 MINMAX-MAXMIN

단계 1. 0에서 N_3로의 최대 흐름량 M을 구한다. $M<\Sigma_{j=1}^{n} p_j$이면 끝난다. 그렇지 않으면 $T \leftarrow N_3$으로 놓는다.

단계 2. LEVEL(T)를 수행하고, 그 결과 과잉 공급구간과 과소 공급구간을 각각 T_0, T_1이라 한다.

단계 3. $T_0=T_1=0$이면 끝난다. 그렇지 않으면 각각 LEVEL(T_0)와 LEVEL(T_1)을 수행하고, 그 결과 과잉 공급구간을 T_{00}, T_{10}, 과소 공급구간을 T_{01}, T_{11}이라 한다.

단계 4. $T_{00}=T_{11}=\phi$이면 끝난다. 그렇지 않으면, T_{00}에서 T_{10}으로, 또 T_{01}에서 T_{11}로 q_i를 재분배 하고, $T_0 \leftarrow T_{00}$, $T_1 \leftarrow T_{11}$으로 놓고, 단계 3으로 간다.

단계 4에서 q_i를 재분배 하는 과정은 LEVEL(T)에서와 마찬가지로 최대 흐름 네트워크 해법을 이용한다.

이 해법의 수행시간을 분석해 보자. 단계 3에서 $|T_0 \cup T_1| - |T_{00} \cup T_{11}| \geq 1$이므로, 단계 3은 최대한 $|T|$회 수행된다. 단계 3이 수행될 때마다 q_i의 재분배를 위해 최대 흐름 네트워크 해법이 사용되나, 이것의 사용 횟수는 $|T|$와 무관하다. 따라서 전체 수행시간은 $0(|T|C(N, E))$로써 MAXMIN 및 MINMAX와 같다.

4. 예 제

하나의 예로써, 3개의 기계로 구성된 병행기계 시스템에서 18개의 작업을 스케쥴링 하는 경우를 생각해 보자. $s_1=3$, $s_2=2$, $s_3=1$이라 할 때, 각 작업의 도착시간, 납기 그리고 작업량이 표 1과 같다고 하자.

이 문제의 네트워크 흐름 모형은, 18개의 작업 node, 84개의 구간-기계 node, 28개의 구간 node 등 131개의 node와 444개의 arc로 구성된다. 여기에서 최대 흐름량은 264가 산출되며, 이것은 $\Sigma_{j=1}^{18} p_j$와 같으므로, 모든 작업이 납기를 지킬수 있는 스케쥴이 가능하다.

가능한 스케쥴을 보장하면서, 각 구간에 할당되는 작업량을 가능한 평준화 하기 위하여, 해법 MINMAX-MAXMIN을 적용하였을 때, 그 결과가 표 2에 나타나 있다. 괄호안의 수치는 어떠한 평준화 해법도 적용하지 않았을 때, 즉 임의의 최대 흐름 네트워크 해법에 의해 가능한 스케쥴이 존재하는 것만 확인했을 때의 결과이다 (여기서는 Dinic[11]의 해법을 사용하였으며, 계산의 편의상 각 arc의 흐름량을 정수로 제한하였다).

표 3은 해법 MINMAX-MAXMIN, MINMAX, MAXMIN을 적용한 각각의 결과와 어

표 1. 각 작업의 도착시간, 납기, 작업량

작업 (j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
도착시간 (r_j)	0	2	5	6	10	12	17	20	20	27	30	35	36	39	41	44	46	50
납기 (d_j)	8	10	15	18	20	22	38	38	40	44	46	50	51	52	55	59	57	60
작업량 (p_j)	10	12	9	13	10	13	15	15	16	14	11	12	17	24	18	22	20	16

표 2. 해법 MINMAX-MAXMIN 에 의한 작업량의 구간별 할당

구간		작업 (i)																	구간별
i	w_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	작업량
1	2	3																		3 (0)
2	3	3(3)	4																	7 (3)
3	1	3(3)	1																	4 (3)
4	2	1(4)	4(6)	2	2															9(10)
5	2		3(6)		6															9 (6)
6	2				3	6														9 (0)
7	3			7(9)	2(5)	1	3													13(14)
8	2				(6)	3(3)	6													9 (9)
9	1				(2)	(3)	1(1)	3												4 (6)
10	2					(4)	3(6)	6												9(10)
11	2						(6)		3	5										8 (6)
12	5							6(7)	5	8										19 (7)
13	3							(8)	5(7)		8									13(16)
14	5								3(9)	1(14)	3(6)	12(1)								19(30)
15	1										(2)	(3)	3(1)							3 (6)
16	2										(2)	(4)	6(6)	3						9(12)
17	1										(1)	(2)	3(3)	2						5 (6)
18	1											(2)	3(3)		2(1)					5 (6)
19	1												1(1)	2(2)	3(3)					6 (6)
20	3												1(3)	4(6)	9(9)	3				17(18)
21	2													1(4)	6(6)	2(2)				9(12)
22	4														4(5)	12(12)	6(6)	1(1)		23(24)
23	1															(1)	2(2)	3(3)		5 (6)
24	1															(1)	2(2)	3(3)		5 (6)
25	3															1(2)	6(6)	9(9)	1(1)	17(18)
26	2																2(2)	4(4)	6(6)	12(12)
27	2																4(4)		6(6)	10(10)
28	1																		3(3)	3 (3)
합계		10	12	9	13	10	13	15	16	14	11	12	17	12	24	18	22	20	16	264

(괄호안의 수치는 평준화 해법을 적용하지 않은 결과임)

떠한 평준화 해법도 적용하지 않은 경과를 비교하고 있다.

표 3에서의 순서대로, q_i/w_i의 표준편차가 0.97, 1.21, 1.09, 1.93로 계산되며, 이로부터 평준화 해법의 효과를 확인할 수 있다. 특히 MINMAX-MAXMIN 에서의 표준편차 값이 가장 적은 것은 기대했던 결과중의 하나이다.

5. 결 론

서로 다른 처리속도를 갖는 **併行機械** 시스템에서, **作業**의 도착시간과 납기가 정해져 있는 경우, 작업량이 작업시간에 따라 보다 평준화된 스케줄을 구하도록 했다. 단위 시간당 작업량이 작업시간의 경과에 따라 가능한 적게 변동하도록

표 3. 해법의 비교

평준화 해법	Minmax-Maxmin		Minmax		Maxmin		적용 않음	
구간	q_i	q_i/w_i	q_i	q_i/w_i	q_i	q_i/w_i	q_i	q_i/w_i
1	3	1.50	0	0.00	3	1.50	0	0.00
2	7	2.33	7	2.33	7	2.33	3	1.00
3	4	4.00	4	4.00	4	4.00	3	3.00
4	9	4.50	9	4.50	9	4.50	10	5.00
5	9	4.50	9	4.50	9	4.50	6	3.00
6	9	4.50	9	4.50	9	4.50	0	0.00
7	13	4.33	13	4.33	13	4.33	14	3.67
8	9	4.50	9	4.50	9	4.50	9	4.50
9	4	4.00	4	4.00	4	4.00	6	6.00
10	9	4.50	9	4.50	9	4.50	10	5.00
11	8	4.00	8	4.00	8	4.00	6	3.00
12	19	3.80	19	3.80	19	3.80	7	1.40
13	13	4.33	13	4.33	13	4.33	15	5.00
14	19	3.80	19	3.80	19	3.80	30	6.00
15	3	3.00	6	6.00	3	3.00	6	6.00
16	9	4.50	9	4.50	9	4.50	12	6.00
17	5	5.00	5	5.00	5	5.00	6	6.00
18	5	5.00	5	5.00	5	5.00	6	6.00
19	6	6.00	6	6.00	6	6.00	6	6.00
20	17	5.67	17	5.67	14	4.67	18	6.00
21	9	4.50	9	4.50	9	4.50	12	6.00
22	23	5.75	23	5.75	23	5.75	24	6.00
23	5	5.00	5	5.00	6	6.00	6	6.00
24	5	5.00	5	5.00	6	6.00	6	6.00
25	17	5.67	17	5.67	18	6.00	18	6.00
26	12	6.00	12	6.00	12	6.00	12	6.00
27	10	5.00	10	5.00	10	5.00	10	5.00
28	3	3.00	3	3.00	3	3.00	3	3.00

하였으며, 이를 위해 단위 시간당 작업량의 최대치와 최소치의 차를 최소화 하는 것을 목표로 하였다. 예제를 통해 이 平準化 해법 MINMAX-MAXMIN의 효과를 확인할 수 있었으며, 기존의 MINMAX, MAXMIN 보다 그 효과가 크다는 것도 알수 있었다.

참고문헌

1. Brown, J.R.: "The Sharing Problem", Operations Research, Vol. 27, No. 1, pp. 324-340, March-April 1979.
2. Federgreun, A., and Groenevelt, H.: "Preemptive Scheduling of Uniform Machines by Ordinary Network Flow Techniques", Management Science, Vol. 32, No. 3, pp.341-349, March 1986.
3. Federgruen, A., and Groenevelt, H.: "Optimal Flow in Networks with Multiple Sources and Sinks, with Applications to Oil and Gas Lease Investment Programs", Operations Research, Vol. 34, No. 2, pp.218-225, March-April 1986.
4. Fujishige, S.: "Lexicographically Optimal Base of a Polymatroid with Respect to a Weight Vector", Mathematics of Operations Research, Vol. 5, pp.186-196, 1980.
5. Gonzalez, T., and Sahni, S.: "Preemptive Scheduling of Uniform Processor Systems", J. of the ACM, Vol. 25, pp.92-101, 1978.
6. Horvath, E., Lam, S., and Sethi, R.: "A Level Algorithm for Preemptive Scheduling", J. of the ACM, Vol. 25, pp.32-43, 1977.
7. Ichimori, T., Ishii, H., and Nishida, T.: "Optimal Sharing", Mathematical Programming, Vol. 23, pp.34-348, 1982.
8. Martel, C.: "Preemptive Scheduling with Release Times, Deadlines, and due Times", J. of the ACM, Vol. 29, No. 3, pp. 812-829, July 1982.

9. Megiddo, N.: "Optimal Flows in Networks with Multiple Source and Sinks", Mathematical Programming, Vol. 7, pp.97-107, 1974.
10. Zeitlin, Z.: "Minimization of Maximum Absolute Deviation in Integers", Discrete Applied Mathematics, Vol. 3, pp. 203-220, 1981.
11. Dinic, E.A.: "Algorithms for Solution of a Problem of Maximum Flow in Networks with Power Estimation", Sov. Math. Doklady, Vol. 11, pp.1277-1280, 1970.