

從屬的인 준비시간을 갖는 多段階工程에서의 그룹스케줄링 解法 (Multi-Stage Group Scheduling with Dependent Setup Time)

宋 永 宇*
魯 仁 珪**
金 滿 植***

Abstract

The application of GT results in the mass production effect to multi-product, small lot-sized production. Group scheduling, which is operations scheduling based on the GT concept, is analyzed in a multistage manufacturing system. But the research has been limited in independent setup time models.

A heuristic algorithm has been developed to minimize the mean flow time when setup times are dependent to sequence of group in multistage.

For evaluation of the heuristic algorithm, the solution of heuristic algorithm is compared with that of random scheduling. The result of comparison indicates that the proposed heuristic algorithm provides good solutions with relatively fewer computational effort.

1. 서 론

다품종 소량생산에서 유사형태 부품이나 유사 공정 부품을 모아 가공하는 그룹 테크놀로지(Group Technology, GT)의 생산방식은 개별생산에서 효율이 높은 흐름생산으로 바뀌 보자

는데 목적이 있으며 최근에 이에 대한 연구가 국내에서도 활발히 진행되고 있다.

GT의 개념을 도입한 단일공정 및 다단계공정의 일정계획인 그룹스케줄링(Group Scheduling, GS)의 이론적 해석 및 최적화를 위한 시도는 1968년 Petrov에 의한 연구 이후로 威. Hit-

* 한양대학교 공과대학 산업공학과 대학원

** 한양대학교 공과대학 산업공학과

*** 한양대학교 공과대학 산업공학과

omi[5] 또는 Hitomi, Nakamura, Yosida[6,8] 등에 의하여 이루어졌고 국내에서는 립[1]의 연구 결과가 발표되었다.

Hitomi, Nakamura 등의 결과는 총소요시간(makespan), 평균처리시간(mean flow time), 순수납기지연(tardiness)의 합을 최소화하기 위해 분기한계법을 이용한 알고리즘의 개발에 초점을 두어 많은 계산량 및 계산시간을 필요로 한다는 문제점을 안고 있다. Yoshida, Hitomi[9]는 그룹의 준비시간이 비독립적인 경우, 단일공정에 대한 분기한계법을 도입하고 있으나 다단계공정에 대해서는 언급치 못하고 있다. 립의 연구에서는 job의 처리순서를 그룹마다 정하고 그에 따라서 그룹순서를 정하는 발견적기법(heuristic method)을 시도하고 있으나 준비 시간이 그룹간 순서에 독립적인 문제에 대해서 순수납기지연의 합을 최소화하기 위한 결과이다. Foo, Wagner[4]는 준비시간이 비독립적인 경우를 고려했으나 단일공정의 총소요시간만을 취급하였다.

본 연구에서는 다단계공정에 있어서 각 그룹의 준비시간은 그룹간 순서에 따라 상이하다는 일반적 전제하에서 이때까지 시도되지 않았던 가공에 있어서의 평균처리시간을 최소화하는 새로운 발견적기법의 개발을 시도한다.

본 연구 모델에 관한 최적화를 구하는 기법이나 발견적방법이 아직 제시되어 있지 않기 때문에 본 알고리즘의 타당성을 평가하기 위해서 수치예를 제시하고 랜덤스케줄링(random scheduling)의 결과와 비교하여 본 알고리즘의 우수성을 검증한다.

2. 그룹스케줄링 모델

2.1 전제조건

그룹생산 방식에 의한 다단계의 기계생산 가공에서 평균처리시간을 최소화하기 위한 그룹스케줄링의 최적화를 목표로 둔 수리모델의 수립

에 있어서 다음의 전제를 설정해 놓는다.

(1) M개의 job이 GT에 의해서 m개의 그룹으로 분류되고 모든 작업의 착수 가능 시점은 동일하다($m < M$)

(2) 생산공정은 ℓ 대의 기계로 구성대고(다단계공정) 각 기계의 고장은 없다.

(3) job의 수행을 위한 준비시간은 각 job의 가공시간(processing time)에 포함되어 있다.

(4) 그룹가공시간(group processing time)은 그룹준비시간(group setup time)과 그룹에 속한 모든 job의 가공시간의 합으로 주어진다.

(5) 그룹준비시간은 그룹순서에 따라 달라진다.

(6) 수행중인 job은 가공이 끝날때까지 기계(공정)에서 제거하지 못한다.(no preemption)

(7) 각 생산공정에서 그룹 및 job의 처리 순서는 전공정을 통해서 동일순서이고, 한 그룹내의 모든 job이 수행된 다음에 다음 순서의 그룹내의 job을 시작할 수 있다.

(8) 본 연구의 스케줄링의 평가기준은 평균처리시간의 최소화이다.

2.2 數理모델의 설정

수리모델 작성에 있어서 M개의 전체job은 가공기술적인 면에서 m개의 그룹으로 분류되며 그룹 i ($G_i; i=1, 2, \dots, m$)에 속하는 n_i 개의 job중 j 번째의 job을 J_{ij} ($i=1, 2, \dots, m; j=1, 2, \dots, n_i$)라고 하며 이들 그룹 및 job의 제k단계 공정($k=1, 2, \dots, \ell$)에서의 기호는 다음과 같다.

P_{ij}^k = 제k단계 공정에서 J_{ij} 의 가공시간

S_{ir}^k = 제k단계 공정에 있어서 G_r 다음에 G_i 가 올 때 G_i 의 준비시간($r=i$ 이면 G_i 가 맨처음에 올때의 준비시간)

P_i^k = 제k단계 공정에서 G_i 의 가공시간. ($= \sum_{j=1}^{n_i} P_{ij}^k$)

Q_i^k = 제k단계 공정에서 G_i 의 생산시간(그룹준비시간+그룹가공시간, $S_{i-1,i}^k + P_i^k$)

$F_{i(t)}^k$ = 생산개시 시점에서 생각하여 제k단계 공정에서 $J_{i(t)}$ 가 완료될 때까지의 소요시간.

$F_{[i]0}^k$ = 생산개시 시점에서 생각하여 제k단계 공정에서 $G_{[i]}$ 의 준비가 완료될 때까지의 소요시간.

단, $\{i\}$ 는 스케줄링에 있어서 그룹 및작업의 처리순서를 나타내는 기호이다. $J_{[i]j}$ 는 i번째로 처리되는 그룹에서 j번째로 처리되는 job을, $G_{[i]}$ 는 i번째로 처리되는 그룹을 의미한다.

제k단계 공정의 $J_{[i]j}$ 에 대해서 작업의 대기상태가 일어나는가 또는 기계의 유휴상태가 일어나는가는 다음과 같은 편차 $\Delta_{[i]j}^k$ 의 부호에 의하여 결정된다.

1) $k = 1$ 인 경우

$$\Delta_{[i]j}^k = 0, \\ W_{[i]j}^k = I_{[i]j}^k = 0$$

2) $k = 2, 3, \dots, \ell$ 인 경우

$$\Delta_{[i]j}^k = F_{[i]j}^{k-1} - F_{[i]j}^k \\ \Delta_{[i]j}^k = \begin{cases} I_{[i]j}^k, & \Delta_{[i]j}^k > 0 \\ -W_{[i]j}^k, & \Delta_{[i]j}^k < 0 \\ 0, & \Delta_{[i]j}^k = 0 \end{cases}$$

$$(i=1, 2, \dots, m; j=1, 2, \dots, n_i)$$

단, $W_{[i]j}^k$ 는 작업의 대기시간을, $I_{[i]j}^k$ 는 기계의 유휴시간을 뜻한다.

$J_{[i]j}$ 가 생산개시 시점에서 전공정을 완료할 때까지 걸리는 시간을 $F_{[i]j}^k$ 라고 하면, 제k단계를 기준으로 $F_{[i]j}^k$ 를 다음과 같이 나타낼 수 있다.

$$F_{[i]j}^k = T_{[0]}^k + \sum_{x=1}^{i-1} (I_{[x]}^k + Q_{[x]}^k) + S_{[i-1]j}^k \\ + \sum_{y=1}^j (I_{[i]y}^k + P_{[i]y}^k) \\ + \sum_{s=k+1}^{\ell} (W_{[i]j}^s + P_{[i]j}^s)$$

$$\text{단, } I_{[x]}^k = \sum_{j=1}^{n_x} I_{[x]j}^k, \quad T_{[0]}^k = \sum_{j=1}^{n_0} S_{[0]j}^k,$$

$$T_{[0]}^1 = 0$$

$$i=1, 2, 3, \dots, m; j=1, 2, \dots, n_i; k=1, 2, \dots, \ell$$

따라서 평균처리시간을 \bar{F} 라고 하면, 제k 단계

를 기준으로 다음과 같이 쓸 수 있다.

$$\bar{F} = (1/M) \sum_{i=1}^m \sum_{j=1}^{n_i} F_{[i]j}^k \\ = (1/M) \sum_{i=1}^m \sum_{j=1}^{n_i} \{T_{[0]}^k + \sum_{x=1}^{i-1} (I_{[x]}^k + Q_{[x]}^k) + S_{[i-1]j}^k + \sum_{y=1}^j (I_{[i]y}^k + P_{[i]y}^k) + \sum_{s=k+1}^{\ell} (W_{[i]j}^s + P_{[i]j}^s)\}$$

문제는 평균처리시간을 최소화하고자 하는 것이므로 문제의 목적함수는 다음과 같다.

$$\text{Min. } \bar{F} = \text{Min}_{a \in A} (1/M) \sum_{i=1}^m \sum_{j=1}^{n_i} F_{[i]j}^k$$

단, a 는 \bar{F} 를 최소화하는 그룹스케줄이고, A 는 $(m!)(\prod_{i=1}^m n_i!)$ 개의 그룹스케줄의 집합이다.

\bar{F} 에 대해서는 총 job수 M 이 일정하므로 \bar{F} 를 최소로 한다는 것은 총처리시간 $F (= \sum_{i=1}^m \sum_{j=1}^{n_i} F_{[i]j}^k)$ 를 최소로 한다는 것과 상응한다.

3. 발견적 그룹스케줄링 기법

그룹스케줄링에서는 각 그룹의 처리순서(그룹간순서)와 각 그룹에 속하는 job의 처리순서(그룹내순서)를 어떻게 결정하는가가 문제이다. 단 일공정에서는 평균처리시간을 평가기준으로 할 때 그룹내순서는 그룹간순서에 영향을 받지 않음을 앞선 연구[8]에 의해서 알 수 있다. 그러나 다단계 공정에서는 일반적으로 그룹내순서는 그룹간순서에 영향을 받고 있다.

본 연구에서는 이러한 영향을 무시하고 그룹간순서와 그룹내순서가 서로 무관하다고 생각하여 그룹간순서와 그룹내순서를 구하게 된다. 그룹간순서는 각 공정에서 종속적인 준비시간을 고려하여 결정하고 그룹내순서는 분기한계법을 사용하여 결정하게 된다.

3-1 그룹간순서를 정하는 방법

$\theta_1 \cup \theta_2 = \theta, \theta_1 \cap \theta_2 = \emptyset$ 를 만족하는 θ 를 m 개의 그룹중에서 이미 순서가 결정된 그룹의 집합을 나타낸다고 하고, θ' 을 아직 순서가 부여되지

않은 그룹의 집합을 나타낸다고 하자. 그러면 θ 는 그룹의 순서가 부여된 집합(an ordered set of groups)으로 간주될 수 있다. $G = \{G_1, G_2, \dots, G_m\}$ 로 정의된 m 개의 그룹의 집합이라면 $\theta \cup \theta' = \theta, \cup \theta, \cup \theta' = G$ 이다. 또한 $|\theta'|$ 와 $|\theta|$ 를 각각 θ 와 θ' 에 속하는 그룹의 수라고 하면 $|\theta| + |\theta'| = |\theta_1| + |\theta_2| + \dots = m$ 이다.

그룹간 순서를 정하기 위해 다음과 같은 기호를 정의하자.

$$\bar{S}_i^k = \sum_{r=1}^m S_{ri}^k / m \quad (k=1, 2, \dots, \ell; i=1, 2, \dots, m; r=1, 2, \dots, m)$$

제 k 단계 공정에서 G_i 의 준비시간 S_{ri}^k 의 합을 전체 그룹수 m 으로 나눈 제 k 단계에서 G_i 의 평균그룹준비시간

$$R_i^k = (\bar{S}_i^k + P_i^k) / n_i \quad (k=1, 2, \dots, \ell; i=1, 2, \dots, m)$$

제 k 단계에서의 평균그룹준비시간과 그룹가공시간의 합을 그룹의 job 수로 나눈 것.

$$R_{.k} = \min_{1 \leq i \leq m} \{R_i^k\}, R_{.k} \in G \quad (k=1, 2, \dots, \ell)$$

제 k 단계에서 가장 작은 R_i^k 의 값을 갖는 그룹

$$\bar{R}_{.k} = \max_{1 \leq i \leq m} \{R_i^k\}, R_{.k} \in G \quad (k=1, 2, \dots, \ell)$$

제 k 단계에서 가장 큰 R_i^k 의 값을 갖는 그룹.

$|G_{.i}| = G_i$ 가 전체 ℓ 개의 공정 중 몇번 $R_{.k}$ 에 속하였는가를 표시하는 횟수

$|\bar{G}_{.i}| = G_i$ 가 전체 ℓ 개의 공정 중 몇번 $\bar{R}_{.k}$ 에 속하였는가를 표시하는 횟수.

단계 1. $\theta_1 = \phi, \theta_2 = \phi, \theta' = \{G_1, G_2, \dots, G_m\}$ 으로 둔다.

단계 2. θ' 에 속한 각 그룹에 대하여 $\bar{S}_i^k, R_i^k, R_{.k}, \bar{R}_{.k}, |G_{.i}|$ 와 $|\bar{G}_{.i}|$ 를 구한다.

단계 3. $\max_i \{|G_{.i}|, i \in \theta'\}$ 인 그룹을 $|\theta_1| + 1$ 번째 그룹순서에 위치시킨다. 단, $\max_i \{|G_{.i}|, i \in \theta'\}$ 인 그룹이 여러개이면 임의로 $|\theta_1| + 1$ 그룹순서에 위치시킨다.

단계 4. $\max_i \{|\bar{G}_{.i}|, i \in \theta'\}$ 인 그룹을 $|\theta_2| + 1$

번째 그룹순서에 위치시킨다. 단, $\max_i \{|\bar{G}_{.i}|, i \in \theta'\}$ 인 그룹이 여러개이면 임의로 $|\theta_2| + 1$ 번째 그룹순서에 위치시킨다.

단계 5. $\max_i \{|G_{.i}|, i \in \theta'\}$ 인 그룹과 $\max_i \{|\bar{G}_{.i}|, i \in \theta'\}$ 인 그룹이 같으면 $|G_{.i}|$ 와 $|\bar{G}_{.i}|$ 를 비교하여 큰 쪽에 따라 배치시키고 같으면 임의로 한다.

단계 6. θ' 에서 θ_1 이나 θ_2 에 들어간 그룹을 제외한다.

단계 7. $|\theta'| > 1$ 이면 단계 2로 가고, $|\theta'| \leq 1$ 이면 단계 8로 간다.

단계 8. $|\theta'| = 1$ 이면 θ' 에 하나 남은 그룹을 $|\theta_1| + 1$ 번째나 $|\theta_2| + 1$ 번째의 그룹순서에 위치시키고 θ_1 의 그룹순서에다 θ_2 의 그룹순서의 역순을 합한다. $|\theta'| = 0$ 이면 그냥 θ_1 의 그룹순서에다 θ_2 의 그룹순서의 역순을 합한다.

이상의 방법을 통하여 그룹간의 순서를 결정할 수 있다.

3.2 그룹내순서를 정하는 방법

그룹의 순서가 정해지면, 임의의 i 그룹에 속하는 job중에서 c 개의 job $J_{i1}, J_{i2}, \dots, J_{ic}$ 를 택하고 이 c 개의 job을 처음부터 $J_{(1)}, J_{(2)}, \dots, J_{(c)}$ 의 순서로 처리할 때의 노우드(node)를 job노우드라고 말하고 $N_c = \{J_{(1)}, J_{(2)}, \dots, J_{(c)}\}$ 로 표시한다. c 개의 job중 여러개의 순서에 따라서 생기는 job노우드의 집합을 작업수준(job level)이라 하고 c 로 표시한다. 이 job노우드에서 분기(branch)는 순서가 결정되어 있지 않은 나머지 $(n_i - c)$ 개의 job중 하나를 $J_{i,c+1}$ 로 표시하고 이 job을 $c + 1$ 번째 처리하는 것으로 하고 job노우드 N_{c+1} 을 작성한다.

하방한계(lower bound)의 계산은 다음과 같이 할 수 있다. 지금 구해진 어느 스케줄에 따라서 생산가공을 하는 경우에 다단계 공정의 제 k 단계 공정이 주공정(critical path)를 형성한

다고 가정한다. 그러면 다단계 공정의 일정계획 문제에서 이 제k단계 공정을 독립된 단일 공정으로 보고 단일공정 일정계획 문제로서 취급할 수 있다. 이러한 공정에서는 l 개(공정 수)가 생각될 수 있으므로 l 개의 각 공정에 대하여 평가 기준치가 계산될 것이다. 따라서 평가 기준치(여기서는 평균처리시간)의 하방한계는 각 job노우드에 있어서 각 공정이 독립된 주공정을 형성하는 것으로 하고 구해진 평가기준치 중에서 그 최대치 $F_c = \max(F_c^1, F_c^2, F_c^3, \dots, F_c^k, F_c^l)$ 에 의하여 구해진다. job 노우드 N_c 에 있어서 총 처리시간 F_c 는 이미 순서가 결정되어 있는 job에 대한 처리시간 F_c^j 와 순서가 결정되어 있지 않은 job에 관한 처리시간 \hat{F}_c^j 에 의하여 다음과 같이 표시된다.

$$F_c = F_c^j + \hat{F}_c^j \quad (1)$$

식(1)에서 제 1 항은 제k단계 공정을 계산의 기준으로 하면 다음과 같다.

$$\begin{aligned} F_c^k &= \sum_{j=1}^c F_{[j]} \\ &= \sum_{j=1}^c \{T_{[0]}^k + \sum_{t=1}^j (I_{[t]}^k + P_{[t]}^k) \\ &\quad + \sum_{s=k+1}^l (W_{[s]}^k + P_{[s]}^k)\} \\ &= cT_{[0]}^k + \sum_{j=1}^c \sum_{t=1}^j (I_{[t]}^k + P_{[t]}^k) \\ &\quad + \sum_{s=k+1}^l \sum_{j=1}^c (W_{[s]}^k + P_{[s]}^k) \end{aligned}$$

$$\text{단, } T_{[0]}^k = F_{[0]}^{k-1}, T_{[0]}^1 = 0, j=1, 2, \dots, n_i; k=2, \dots, l$$

$F_{[j]}^k$ = 생산개시 시점에서 생각하여 제k단계 공정에서 $J_{[j]}$ 가 완료될 때까지의 소요시간.

$\Delta_{[j]}^k = F_{[j]}^k - F_{[j-1]}^k$ 라고 정의할 때 $I_{[j]}^k$ 와 $W_{[j]}^k$ 는 다음조건을 만족하는 기계유휴시간과 작업의 대기시간이다.

$$\begin{aligned} I_{[j]}^k, \Delta_{[j]}^k &> 0, \quad k=2, 3, \dots, l \\ \Delta_{[j]}^k &= \begin{cases} -W_{[j]}^k, & \Delta_{[j]}^k < 0, \quad k=2, 3, \dots, l \\ 0, & \Delta_{[j]}^k = 0, \quad k=1, 2, 3, \dots, l \end{cases} \end{aligned}$$

그리고 $J_{[j]}$ 가 생산개시 시점에서 전공정을 완

료할 때까지 걸리는 시간(처리시간)을 $F_{[j]}$ 라 하고, $P_{[j]}^k$ 는 제k단계 공정에서 $J_{[j]}$ 의 가공 시간이다.

식(1)의 제 2 항에 대해서는 그 하방한계 \hat{F}_c^j 를 다음과 같이 설정한다.

$$\begin{aligned} \hat{F}_c^j &= \sum_{j=c+1}^{n_i} \hat{F}_{[j]} \\ &= \sum_{j=c+1}^{n_i} (F_{[0]}^k + \sum_{t=c+1}^j P_{[t]}^k + \sum_{s=k+1}^l P_{[s]}^k) \\ &= (n_i - c)F_{[0]}^k + \sum_{j=c+1}^{n_i} \sum_{t=c+1}^j P_{[t]}^k \\ &\quad + \sum_{s=k+1}^l \sum_{j=c+1}^{n_i} P_{[s]}^k \end{aligned}$$

단, $F_{[0]}^k = 0$ 이고 두번째 항은 작은 가공치를 가지는 job의 순서로 처리한다. 즉, $P_{[c+1]}^k \leq P_{[c+2]}^k \leq \dots \leq P_{[n_i]}^k$ 의 순서로 처리한다.

결국 job노우드 N_c 에 있어서 job에 대한 총 처리시간의 하방한계는 다음과 같이 구해진다.

$$\hat{F}_c = \max(F_c^j + \hat{F}_c^j, F_c^k + \hat{F}_c^k, \dots, F_c^k + \hat{F}_c^k, F_c^l + \hat{F}_c^l)$$

그런데 $F_c^j = F_c^2 = \dots = F_c^k = \dots = F_c^l = F_c^j$ 이므로 결국 $\hat{F}_c = F_c^j + \max(\hat{F}_c^j, \hat{F}_c^k, \dots, \hat{F}_c^k, \dots, \hat{F}_c^l)$ 이다.

단계 1. 임의의 그룹 i 에 대해서 $c=1$ 이면 job을 분기해서 job 노우드 N_c 를 만든다.

단계 2. 각 job노우드 N_c 에 대한 하방한계 \hat{F}_c 를 계산하고 $\hat{F} = \min(\hat{F}_c)$ 인 job에 대하여 분기한다.

단계 3. $c=c+1$ 로 하고 각 노우드의 \hat{F}_c 를 계산한다. $c=1, 2, \dots, c$ 에 대하여 아직 분기되지 않은 모든 노우드에 대한 $\hat{F} = \min(\hat{F}_c)$ 를 구하고 \hat{F} 의 값을 갖는 노우드를 분기한다.

단계 4. $c < n_i + 1$ 이면 단계 3으로 가고 $c = n_i + 1$ 이면 멈춘다. 이때 $c = n_i$ 일때 최종 분기된 job노우드가 G_i 의 최적 job 스케들들이 된다.

3.3 수치예제

3 단계의 공정을 거치는 3 개의 그룹 G_1, G_2, G_3 에 대하여 Table 1. 같은 각 job 당 각 공정에서의 가공시간이 나와 있고, 각 공정마다 그룹간 순서에 따라 달라지는 준비시간이 Table 2. 와 같이 주어졌다. 문제는 평균처리시간을 최소화하는 최적해에 비교적 가까운 근사해를 구하는 것이다.

Table 1. Processing times for a three job, three-group and three-stage flow shop scheduling problem.

Stage	G_1			G_2			G_3		
	J_{11}	J_{12}	J_{13}	J_{21}	J_{22}	J_{23}	J_{31}	J_{32}	J_{33}
1	3	2	7	2	2	7	4	6	6
2	1	1	7	6	6	9	6	9	3
3	4	8	2	7	9	9	3	6	2

Table 2. Setup times between three groups in three stage.

Stage Group		1			2			3		
r	i	1	2	3	1	2	3	1	2	3
1	1	2	8	8	4	9	6	9	4	8
2	1	4	6	9	1	9	9	6	2	3
3	1	1	9	8	6	5	8	8	6	5

The element of table is S_{ri}^ .

각 원소는 S_{ri}^* 를 나타내고 있으며, $r=i$ 면 공정의 맨앞에 G_i 가 올때의 준비시간이다.

(그룹간 순서)

단계 1. $\theta' = \{G_1, G_2, G_3\}, \theta_1 = \theta_2 = \phi$

단계 2. $\bar{S}_1^1 = \frac{1}{3}(2+4+1) = 2.33,$
 $\bar{S}_1^2 = \frac{1}{3}(4+1+6) = 3.67,$
 $\bar{S}_1^3 = \frac{1}{3}(9+6+8) = 7.67,$
 $\bar{S}_2^1 = 7.67, \bar{S}_2^2 = 7.67, \bar{S}_2^3 = 4,$
 $\bar{S}_3^1 = 8.33, \bar{S}_3^2 = 7.67, \bar{S}_3^3 = 5.33$

$R_1^1 = \frac{1}{3}(2.33+12) = 4.77,$

$R_1^2 = \frac{1}{3}(3.67+9) = 4.22,$

$R_1^3 = \frac{1}{3}(7.67+14) = 7.22,$

$R_2^1 = 6.22, R_2^2 = 9.55, R_2^3 = 9.66,$

$R_3^1 = 8.11, R_3^2 = 8.55, R_3^3 = 5.44,$

$R_{1,1} = \{G_1\}, R_{1,2} = \{G_2\}, R_{1,3} = \{G_3\},$

$\bar{R}_{1,1} = \{G_3\}, \bar{R}_{1,2} = \{G_2\}, \bar{R}_{1,3} = \{G_1\}$

$\therefore |G_{1,1}| = 2, |G_{1,2}| = 0, |G_{1,3}| = 1,$
 $|\bar{G}_{1,1}| = 0, |\bar{G}_{1,2}| = 2, |\bar{G}_{1,3}| = 1$

단계 3. $\max_i \{ |G_{i,1}| \} = G_1 \quad \therefore \theta_1 = \{G_1\}$

단계 4. $\max_i \{ |\bar{G}_{i,1}| \} = G_2 \quad \therefore \theta_2 = \{G_2\}$

단계 6. $\theta' = \{G_3\}$

단계 7. $|\theta'| = 1$

단계 8. $\theta = \{G_1, G_2, G_3\}$

이상의 그룹간순서를 정하는 단계를 통하여 그룹간의 처리순서가 정해졌다.

(그룹내 순서)

앞에서 설명한 분기한계법을 사용하면 각 그룹마다 다음과 같은 결과를 얻을 수 있다.

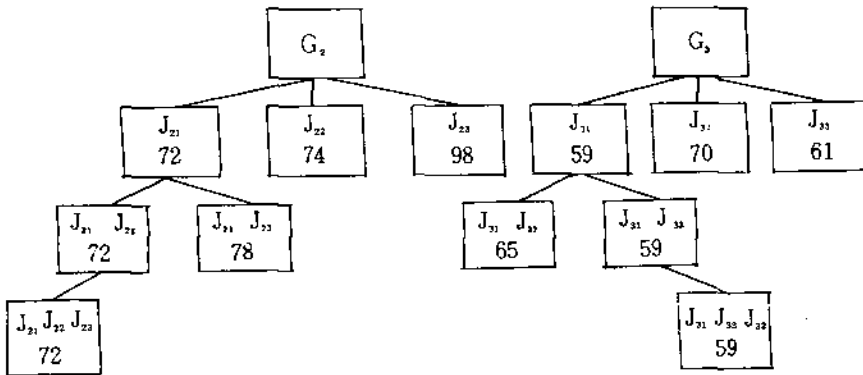
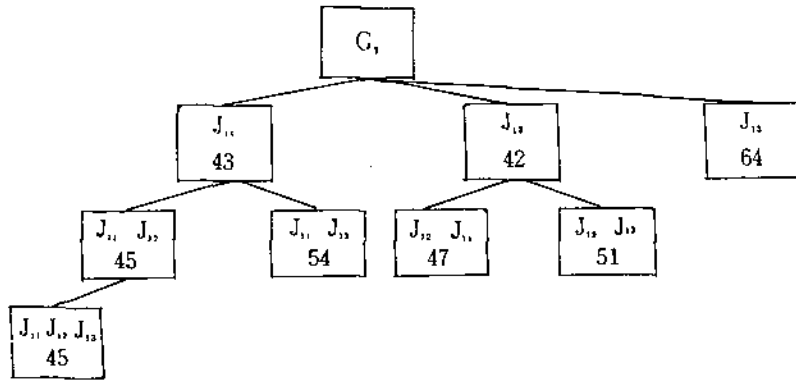


Figure 1. Calculation of Branch and Bound

결론적으로 발견적 기법에 의한 문제의 그룹 스케줄링은 Table 3. 와 같다.

이것을 간트 도표로 나타내면 Figure 2. 와 같다.

Table 3. Heuristic solution of example

group schedule	G ₁	G ₂	G ₃
job schedule	J ₁₁ J ₁₂ J ₁₃	J ₃₁ J ₃₂ J ₁₃	J ₁₁ J ₂₂ J ₂₃

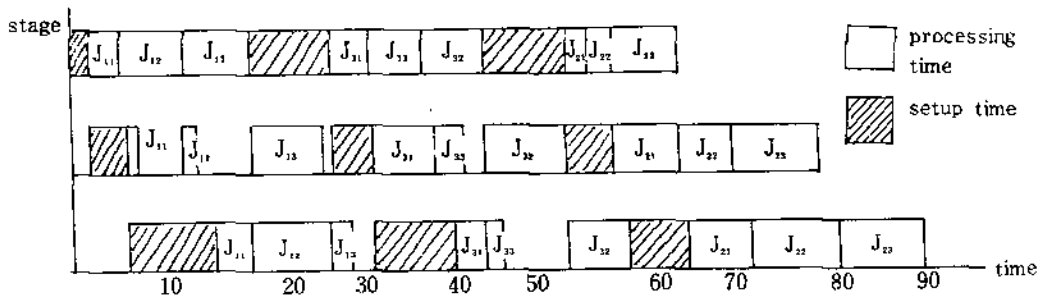


Figure 2. Gantt Chart of Group Scheduling

4. 발견전 그룹스케줄링 기법의 평가

일반적으로 실제적 문제에 대한 해를 간단히 빨리 얻기 위해서 우수한 발견적기법의 필요성이 증가되고 있으나, 그 유효성을 평가하는 일반적인 방법론이 없다.

본 연구에서는 발견적기법의 유효성을 평가하기 위해서 랜덤스케줄링에 의해 얻은 해와 발견적기법에 의해 얻은 해를 비교했다.

시험을 위한 그룹스케줄링 문제를 단계(공정 또는 기계)수 3에서 5까지 변화시키며 각각 3, 4, 5개의 그룹을 갖고 그룹당 job수가 3에서 9까지 되도록 만들었다. job 가공시간이나 그룹준비시간은 VAX-11/780컴퓨터에 내장된 난수발생기(random number generator)를 사용하여 1에서 9까지의 정수로 랜덤하게 만들었다.

시험한 문제는 각 경우마다 1문제씩 64개의 문제를 반복하여 총 126개이며, 본 연구에서 개

발한 발견적기법을 FORTRAN으로 프로그램 하고 VAX-11/780컴퓨터를 사용하여 각 문제당 1000개의 랜덤스케줄을 만들어 평가하였다.

4.1 발생빈도에 따른 평가

평가의 척도로 발견적기법에 의해서 얻은 해로 인한 평균처리시간이 랜덤스케줄링에 의해서 얻어진 해로 구한 평균처리시간보다 더 작게 나타나는 발생 빈도를 고려하였다.

Table 4.는 발견적기법에 의해 얻은 그룹스케줄이 각 문제당 1000개의 랜덤스케줄보다 우수한 발생빈도를 나타낸다. Table 4.를 보면 시험한 126개의 문제중 19.8%인 25문제에서 100%의 발생빈도로 랜덤스케줄보다 우수한 근사 최적해를 얻을 수 있었고 평균적으로 86.88%의 발생 빈도로 근사최적해가 랜덤스케줄보다 우수 하였다.

Table 4. Comparison of frequency between heuristic solution and random solution

frequency (%)	number of problems	percentage of problems (%)	cumulative percentage of problems (%)
100.00	25	19.8	19.8
99.95-89.95	53	42.1	61.9
89.95-79.95	15	11.9	73.8
79.95-69.95	12	9.5	83.3
69.95-59.95	8	6.3	89.6
59.95-49.95	6	4.8	94.4
49.95-39.95	2	1.6	96.0
39.95-29.95	4	3.2	99.2
29.95-19.95	1	0.8	100.0
19.95-9.95	0	0.0	
9.95-0.00	0	0.	
total	126	100.0	

이 시험과정에서 발견적 기법의 유효성과 문제의 매개변수(그룹수, 그룹내 job수, 공정수) 사이에 어떠한 관계가 있는가 알아보기 위해서

분산분석(ANOVA)을 행한 것이 Table 5. 에 나와 있다.

Table 5. ANOVA for factors of heuristic method

source of variation	sum of squares	df	mean squares	F
stage	243. 573	2	121. 787	0. 415
group	1179. 146	2	589. 573	2. 008
job	5125. 078	6	854. 180	2. 909*
stage x group	422. 047	4	105. 512	0. 359
stage x job	4772. 977	12	397. 748	1. 355
group x job	2763. 776	12	230. 315	0. 784
stage x group x job	6288. 767	24	262. 032	0. 892
residual	18498. 428	63	293. 626	
total	39293. 793	125		

* : a case that is significant with $\alpha=0.05$

분산분석 결과 주요인중 그룹내의 job수가 유의수준 $\alpha=0.05$ 로 발생빈도에 영향을 미침을 알 수 있다. 즉 그룹내의 job수가 변함에 따라 랜덤 스케줄보다 우수한 근사최적해의 발생빈도가 변함을 알 수 있는데 그룹내의 job수의 변화에 따른

발생빈도의 변화를 산포도(scattergram)로 나타내면 Figure 3와 같게 된다. 산포도를 통해서 그룹내 job 수가 늘어남에 따라 발생빈도가 좋아짐을 알 수 있다.

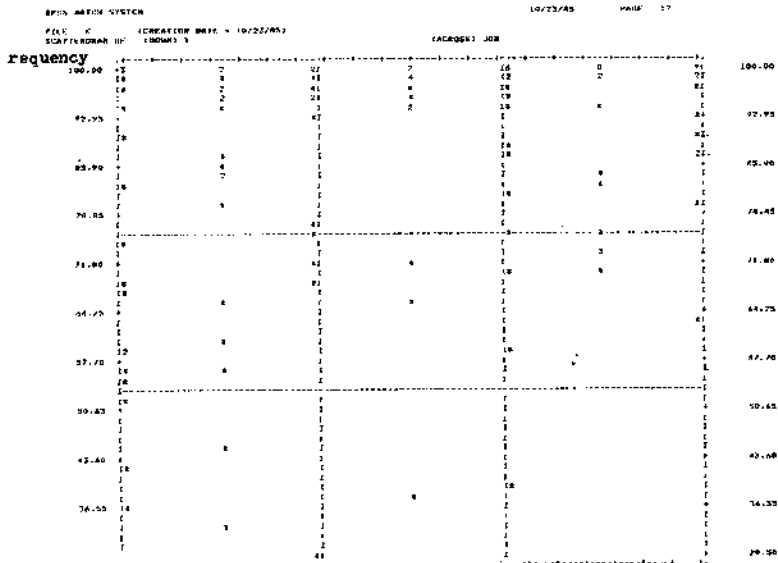


Figure 3. Scattergram of frequency according to number of jobs.

4.2 단축비에 의한 평가

발견적기법에 의해서 얻어진 해가 랜덤스케줄링에 의해서 얻어진 해보다 얼마큼 평가기준(본 연구에서는 평균처리시간)을 단축시키는데 고려하기 위해서 평균단축비를 평가의 척도로 생각할 수 있다. 단축비를 구하기 위해 다음과 같은 기호를 정의하자.

\bar{F}_r = 1000개 랜덤스케줄링에 의해서 얻어진 평균처리시간의 평균으로 랜덤스케줄링에 의한 평균처리시간을 대표.

\bar{F}_h = 발견적기법에 의해서 얻어진 평균 처리시간.

단축비 R은 다음과 같다.

$$R = (\bar{F}_r - \bar{F}_h) / \bar{F}_h \times 100 (\%)$$

R이 음수이면 발견적 기법의 해가 평균 처리시간을 증가시켜 랜덤스케줄링보다 더 나쁜 결과를 가져온 것이고, R이 양수이고 큰 값을 가질수록 랜덤스케줄링에 의한 평균 처리시간을 발견적기법에 의해서 R만큼 단축시켰다는 것을 알 수 있다.

각 문제 형태에 따른 평균단축비는 Table 6에

나와 있다.

Table 6. Reduction ratio according to change of numbers of groups and jobs(%)

job \ group	3	4	5
3	8	2	4
4	5	7	9
5	9	9	5
6	7	9	10
7	8	5	7
8	6	7	7
9	10	6	8

Table 6.은 각 경우의 문제를 6개씩 평가한 평균으로 총 126문제의 평균단축비 \bar{R} 는 7.0%임을 알 수 있다. 즉 발견적기법에 의해서 얻어진 스케줄의 평균처리시간은 랜덤스케줄링에 의해서 구한 평균처리시간을 7.0% 단축시켰음을 알 수 있다. 그리고 단축비 R은 그룹수나 job수 또는 단계(공정)수의 변화에 큰 영향을 받고 있지 않음을 연구 결과 알 수 있었다.

4.3 소요계산시간

Table 7.은 126 문제를 푸는데 소요되는 시간

Table 7. CPU Time of heuristic method(sec)

stage	group	job						
		3	4	5	6	7	8	9
3	3	0.83	0.80	0.83	0.96	1.12	1.73	1.84
	4	0.99	1.02	1.10	1.35	1.67	2.97	3.49
	5	1.30	1.33	1.30	1.52	1.88	2.68	4.48
4	3	0.85	0.79	0.88	1.01	1.49	1.80	3.19
	4	0.93	1.01	1.18	1.44	1.69	2.47	5.08
	5	1.24	1.32	1.40	1.82	2.17	3.21	7.04
5	3	0.87	0.81	0.90	1.13	1.59	2.33	4.21
	4	1.04	1.07	1.23	1.58	2.14	3.99	5.54
	5	1.26	1.48	1.59	2.09	3.29	4.39	8.88

을 그룹과 job 그리고 공정 의 크기에 따라서 표시해 놓은 것이다. 이 시간은 VAX-11/780 컴퓨터로 각 경우의 문제를 2 개씩 랜덤하게 만들어 발견적 기법으로 문제의 해를 구하는데 걸리는 CPU Time 의 평균을 나타낸 것이다. 연구 결과 그룹수와 job 수 그리고 공정수가 CPU Time 의 변화에 큰 영향을 미치고 있음을 알 수 있었다.

5. 결 론

GT의 개념을 도입한 그룹스케줄링 모델을 다

품종, 다단계 생산시스템에서 준비시간이 그룹 순서에 증속적인 경우에 평균처리시간을 평가기준으로 분석 및 고찰하였다. 문제에 접근하는 방법으로 발견적 기법을 개발하였고 그 유효성을 평가하기 위해 랜덤스케줄링과 비교하였다.

분석한 결과 총 126문제에 대하여 발견적기법에 의해 얻어진 해가 각 문제당 1000개의 랜덤스케줄링에 의해 구해지는 해보다 약 87%의 우수한 발생빈도를 가짐을 알 수 있었고, 랜덤스케줄링에 의해서 구한 평균처리시간을 7% 단축시킴을 알 수 있었다.

References

1. 조 규갑, "다단계 생산시스템에서의 그룹스케줄링에 대한 연구", 대한산업공학회지, 9권, 1호, pp 23-31, 1983.
2. Baker, K.R., Introduction to Sequencing and Scheduling, Wiley, New York, 1974.
3. Ham, I., Hitomi, K., and Yoshida, T., Group Technology, Kluwer-Nijhoff Publishing, 1985.
4. Foo, F.C., and Wager, J.G., "Set-up Times in Cyclic and Acyclic Group Technology Scheduling Systems," Int. J. Pro. Res., Vol. 21, pp. 63-73, 1983.
5. Hitomi, K., and Ham, I., "Group Scheduling Technique for Multiproduct, Multistage Manufacturing Systems," Transactions of the American Society of Mechanical Engineers, Vol. 99, pp. 759-765, 1977.
6. Nakamura, N., and Hitomi, K., "Analysis of Production Scheduling (Optimization of Group Scheduling on Multiple Production Systems)," (Japanese), Transactions of the Japan Society of Mechanical Engineers, Vol. 42, pp. 2964-2975, 1976.
7. Ravisethi, "On the Complexity of Mean Flow Time Scheduling," Mathematics of Operations Research, Vol. 2, pp. 320-330, 1977.
8. Yosida, T., Nakamura, N., and Hitomi, K., "Optimization of Group Scheduling for a Single Stage Production," (Japanese), Transactions Japan Society of Mechanical Engineers, Vol. 39, pp. 1933-2003, 1973.
9. Yosida, T., Hitomi, K., "Optimization of Group Scheduling for a Single Stage Production with Dependent Setup Time," (Japanese), Transactions Japan Society of Mechanical Engineers, Vol. 43, pp. 1531-1538, 1977.