Journal of the Korean Institute of Industrial Engineers
Vol. 9. No. 1. June 1983

多段階生産시스템에서의 그룹스케듈링에 대한 硏究

(A Study of Group Scheduling in Multi-Stage Manufacturing Systems)

曺 圭 甲*

Abstract

A group scheduling problem, which is production scheduling problem associated with the concept of group technology, is studied under due date constraints in multi-stage manufacturing systems. The purpose of this paper is to develop and evaluate a practical heuristic procedure for determining group sequence and job sequence within each group to minimize total tardiness in multi-stage manufacturing systems.

A heuristic algorithm has been developed by introducing the concept of relative measures of job tardiness and group tardiness for job sequencing and group sequencing, respectively. A numerical example is shown to illustrate the proposed procedure.

The heuristic algoirthm is tested by comparisons with problems with known optimal solutions and also with random group schedules for a set of large-size problems. . Results indicate that the proposed heuristic algorithm provides good solutions with small computational requirements, and thus is viable for large-size problems in practice.

1. 序 論

그룹 테크노로지(Group Technology, GT)의 개념이 多種中少量生産의 生産性向上을 위해서 생산 시스템의 여러 분야에 도입되었다. 그 가운데 생산스케듈링이 *GT*의 개념을 도입한 생산스케듈링을 '그룹 스케듈링 (Group Scheduling, GS)'이라 칭한다[1]. *GT*의 개념에 의해 유사한 부품들이 그룹화되어 차례대로 가공되므로, *GS*는 그룹간의 순서결정과 각 그룹내의 부품의 가공순서의 결정에 관한 두가지 문제에 대한 최적스케듈링을 다루는 2단계 생산스케듈링이라고 볼 수 있다.

*GS*는 일반적인 생산스케듈링과 다른 여러가지 특성을 가지고 있으며[2], 그중 가장 중요한 것은 유사한 부품들이 그룹화되어 차례대로 가공되므로 각 부품 가공의 준비시간(setup time)은 감소되거나 제거될 수 있고 단지 각 그룹에 대한 준비시간만이 소요된다는 것이다. 이것은 다음 표1과 2를 비교해 보면 쉽게 알 수 있다. 표2에서 각 부품가공의 준비시간은 그

* 釜山大學校 工科大學

부품들이 소속하는 그룹의 준비시간에 비해 아주 작다 고 볼 수 있기 때문에 부품(또는 job)의 준비시간은 부품의 가공시간에 포함되었다고 가정한다.

Table 1. A Representation of Data for a Flow-type Production Scheduling Problem

Job		J_1		J_2		J_3		......	J_j		......	J_{N-1}		J_N	
Setup/ Processing Time		S_1^k	P_1^k	S_2^k	P_2^k	S_3^k	P_3^k	......	S_j^k	P_j^k	......	S_{N-1}^k	P_{N-1}^k	S_N^k	P_N^k
MACHINE	1	S_1^1	P_1^1	S_2^1	P_2^1	S_3^1	P_3^1	......	S_j^1	P_j^1	......	S_{N-1}^1	P_{N-1}^1	S_N^1	P_N^1
	2	S_1^2	P_1^2	S_2^2	P_2^2	S_3^2	P_3^2	......	P_j^2	P_j^2	......	S_{N-1}^2	P_{N-1}^2	S_N^2	P_N^2
	3	S_1^3	P_1^3	S_2^3	P_2^3	S_3^3	P_3^3	......	S_j^3	P_j^3	......	S_{N-1}^3	P_{N-1}^3	S_N^3	P_N^3
	.	.	.	.	.	.	.	......	.	.	......	.	.	.	.
	.	.	.	.	.	.	.	......	.	.	......	.	.	.	.
	.	.	.	.	.	.	.	......	.	.	......	.	.	.	.
	K	S_1^K	P_1^K	S_2^K	P_2^K	S_3^K	P_3^K	......	S_j^K	P_j^K	......	S_{N-1}^K	P_{N-1}^K	S_N^K	p_N^K
Due Date		d_1		d_2		d_3		......	d_j		......	d_{N-1}		d_N	

Note: (1) S_j^k : Setup time of job j on machine k (j=1, 2, ……, N, k=1, 2, 3, ……, K)

(2) P_j^k : Processing time of job j on machine k

(3) d_j : Due date of job j

Table 2. A Representation of Data for a Group Scheduling Problem

Group		G_1					G_2				......	G_M			
Job		–	J_{11}	J_{12}	......	J_{1n_1}	–	J_{21}	......	J_{2n_2}	......	–	J_{M1}	......	J_{Mn_M}
Setup / Processing Time		S_1^k	P_{11}^k	P_{12}^k	......	$P_{1n_1}^k$	S_2^k	P_{21}^k	......	$P_{2n_2}^k$	......	S_M^k	P_{M1}^k	......	$P_{Mn_M}^k$
MACHINE	1	S_1^1	P_{11}^1	P_{12}^1	......	$P_{1n_1}^1$	S_2^1	P_{21}^1	......	$P_{2n_2}^1$	......	S_M^1	P_{M1}^1	......	$P_{Mn_M}^1$
	2	S_1^2	P_{11}^2	P_{12}^2	......	$P_{1n_1}^2$	S_2^2	P_{21}^2	......	$P_{2n_2}^2$	......	S_M^2	P_{M1}^2	......	$P_{Mn_M}^2$
	3	S_1^3	P_{11}^3	P_{12}^3	......	$P_{1n_1}^3$	S_2^3	P_{21}^3	......	$P_{2n_2}^3$	......	S_M^3	P_{M1}^3	......	$P_{Mn_M}^3$
	.	.	.	.	......	.	.	.	......	.	......	.	.	......	.
	.	.	.	.	......	.	.	.	......	.	......	.	.	......	.
	.														.
	.	.	.	.	......	.	.	.	......	.	......	.	.	......	.
	K	S_1^K	P_{11}^K	P_{12}^K	......	$P_{1n_1}^K$	S_2^K	P_{21}^K	......	$P_{2n_2}^K$	......	S_M^K	P_{M1}^K	......	$P_{Mn_M}^K$
Due Date		–	d_{11}	d_{12}	......	d_{1n_1}	–	d_{21}	......	d_{2n_2}	......	–	d_{M1}	......	d_{Mn_M}

Note: (1) Total number of job $N=\sum_{i=1}^{M} n_i$, where n_i indicates number of jobs in group i (i=1, 2, ……, M)

(2) S_i^k : Setup time of group i on machine k (k=1, 2, ……, K)

(3) P_{ij}^k : Processing time of job j of group i on machine k (j=1, 2, ……, n_i)

(4) d_{ij} : Due date of job j of group i

單一工程 및 多段階工程에 대한 GS 모델들을 몇가지 스케듈링 평가기준하에 분석하여 그룹순서및 job수

서를 결정하는 최적화 방법 및 발견적 방법이 제안되어 있다[1－7]. 생산스케듈링에서 스케듈을 평가하는 여러가지의 평가기준이 있는데[8], 그중에서 중요하고 어려운 문제의 하나는 납기(due date)를 맞추는 생산스케듈링문제이다. 납기에 관련되는 평가기준중에 중요한 것이 순수납기지연(tardiness)에 관계되는 것이다. GS 에서 단일공정및 다단계공정에 대한 순수납기지연의 合을 최소화하는 최적화 방법이 제안되었는데[4, 6], 이 두가지 방법은 分岐限界法을 이용했기 때문에 소요계산량과 계산시간이 많이 소요된다는 문제점이 있다.

순수납기지연의 합을 최소화하는 문제는 NP-complete로 알려진 상당히 해결하기 어려운 組合의 문제이다. 이같이 해결하기 어려운 組合問題의 복잡성과 난해성때문에 실제로 쉽게 사용할 수 있는 發見的 技法의 개발에 대한 관심이 증가되고 있다[9]. GS 에서 순수납기지연의 합을 최소화하는 실용적 방법이 개발되어 있지 않으므로 이 방법의 개발이 필요하다고 사료된다.

본 연구에서는 다단계생산시스템에서 순수납가지연의 합을 최소화하는 GS 문제에 대한 분석 및 근사최적해를 얻는 방법을 제안하고 또 그 유효성을 평가하고자 한다.

2. 多段階工程에서의 그룹스케듈링

2.1 前提條件

다단계생산시스템에서 납기제약조건하의 GS 문제에 대한 수학적 모델을 만들기 위해서 다음의 전제조건을 설정한다.

(1) N個의 部品(또는 job)이 GT에 의해서 M個(단, $M<<N$)의 部品群으로 분류되고, 모든 部品은 시간 0에서 가공이 가능하다.

(2) 생산공정은 다단계로서 K대의 기계로 구성되고, 각 기계의 고장은 없다고 가정한다.

(3) 부품의 가공을 위한 준비시간은 부품의 가공시간(Processing time)에 포함되어 있다고 가정한다.

(4) 그룹가공시간(group processing time)은 그룹준비시간(group setup time)과 그 그룹에 속한 모든 부품의 가공시간의 합으로 주어지며, 그룹준비시간은 그룹순서에 대해서 독립적이라 가정한다.

(5) 가공중인 부품은 가공이 끝날때까지 기계에서 제거하지 못한다(no preemption).

(6) 모든 기계에서 그룹 및 job의 순서는 동일하고(no passing), 한 그룹내의 모든 부품이 가공된 다음에 다음 순서의 그룹내의 부품의 가공을 시작할 수 있다.

(7) 각 기계에서 그룹준비시간 및 부품의 가공시간, 그리고 부품의 납기가 주어져 있다.

(8) 스케듈의 평가기준은 순수납기지연의 합계(total tardiness)이다.

이상의 전제조건하에서 보면 주어진 문제는 確定的이고 靜的인 多段階 그룹스케듈링문제에서 모든 job의 순수납기지연의 합계를 최소화하는 문제이다.

2.2 問題의 數學的 模型

다단계공정에서 순수납기지연의 합을 최소화하는 GS 문제의 분석을 위해서 다음의 기호를 정의한다.

i =그룹 인덱스($i=1, 2, \cdots\cdots, M$)

j =job 인덱스($j=1, 2, \cdots\cdots, n_i$; 단, n_i 는 그룹 i에 속하는 job의 수)

k =공정(또는 기계) 인덱스($k=1, 2, \cdots\cdots, K$)

M_k =기계 k

G_i =그룹 i

J_{ij} =그룹 i의 job j

p_{ij}^k =공정k에서의 J_{ij} 의 가공시간

S_i^k =공정k에서의 G_i 의 준비시간

Q_i^k =G_i 의 기계 k에서의 그룹가공시간

d_{ij} =J_{ij} 의 납기

그룹 i의 기계 k에서의 그룹가공시간은

$$Q_i^k = S_i^k + \sum_{j=1}^{n_i} P_{ij}^k \quad (i=1, 2, \cdots, M;\ k=1, 2, \cdots, K) \qquad (1)$$

로 주어진다.

부호 [ ]는 그룹스케듈에서 그룹과 Job의 위치를 나타내기 위해 사용한다. 즉 $J_{[i,j]}$는 그룹스케듈에서 i번째 그룹의 j번째 job을 의미하고 $G_{[i]}$는 i번째 그룹을 의미한다.

$J_{[i,j]}$의 M_k에서 가공완료시간 $C_{[i,j]}^k$는

$$C_{[i,j]}^k = \sum_{l=1}^{i-1} \left(\sum_{j=1}^{n_l} w_{[l,j]}^k + Q_{[l]}^k \right) + S_{[i]}^k + \sum_{m=1}^{j} \left(w_{[i,m]}^k + p_{[i,m]}^k \right) \qquad (2)$$

로 주어진다. 식(2)에서 $w_{[i,j]}^k$는 M_k에서 $G_{[i]}$의 ($j-1$)번째 부품을 가공한 뒤 j번째 부품을 가공하기 전까지의 대기 시간을 나타내며, [j]=1에 대해서

$$w_{[i,1]}^k = \begin{cases} C_{[i,1]}^{k-1} - C_{[i-1,n_{i-1}]}^k - S_i^k, & C_{[i,1]}^{k-1} > C_{[i-1,n_{i-1}]}^k + S_{[i]}^k \\ 0, & \text{나머지 경우} \end{cases} \qquad (3)$$

이고, [j]≠1에 대해서는

$$w^k_{[i,j]}=\begin{cases} C^{k-1}_{[i,j]}-C^k_{[i,j-1]}, & C^{k-1}_{[i,j]}>C^k_{[i,j-1]} \\ 0, & \text{나머지 경우}\end{cases} \quad (4)$$

로 주어진다.

$J_{[i,j]}$의 가공완료시간 $C_{[i,j]}$는

$$C_{[i,j]}=max\ (C^k_{[i,j-1]},\ C^{k-1}_{[i,j]})+p^k_{[i,j]} \quad (5)$$

이고, $J_{[i,j]}$의 순수납기지연 $T_{[i,j]}$는

$$T_{[i,j]}=max\ (0,\ C_{[i,j]}-d_{[i,j]}) \quad (6)$$

이며, 여기서 $d_{[i,j]}$는 $J_{[i,j]}$의 납기이다.

따라서 모든 job의 순수납기지연의 합계 T는

$$T=\sum_{i=1}^{M}\sum_{j=1}^{n_i} T_{[i,j]} =\sum_{i=1}^{M}\sum_{j=1}^{n_i}\{max\ (0,\ C_{[i,j]}-d_{[i,j]})\} \quad (7)$$

로 주어지며, 이 T를 최소화하고자 하는 것이다.

앞에서 도입한 전제조건에 따른 주어진 문제의 목적함수는 다음과 같다[12].

$$min\ T=\min_{s\in S}\{\sum_{i=1}^{M}\sum_{j=1}^{n_i} f(C_{ij})\}$$

단, $s=T$를 최소화하는 그룹스케듈

$S=M!\ \pi^M_{i=1}(n_i\ !)$개의 그룹스케듈의 집합

$C_{ij}=J_{ij}$의 가공완료시간

$f\ (C_{ij})=max\ (0\ ,\ C_{ij}-d_{ij})$

3. 発見的 그룹스케듈링 技法[10]

3.1 예비적 고찰

σ를 M個의 그룹중에서 이미 순서가 결정된 그룹의 집합을 나타내고, σ'을 아직 순서가 부여되지 않은 그룹의 집합을 나타낸다고 한다. 따라서 σ는 그룹의 순서가 부여된 집합(an ordered set of groups) 으로 간주할 수 있다. $G=\{G_1, G_2, \cdots\cdots, G_M\}$로 정의된 M개의 그룹의 집합이라면, $\sigma U\sigma'=G$이다. 또 $|\sigma|$와 $|\sigma'|$을 각각 σ와 σ'에 속하는 그룹의 수라고 하면 $|\sigma|+|\sigma'|=M$이다.

q_σ를 σ에 속하는 마지막 그룹 G_n의 마지막 job의 가공완료시간을 나타낸다면,

$$q_\sigma=\sum_{\substack{i=1\\ i=i\in\sigma}}^{n} (W^K_{[i]}+Q^K_{[i]}) \quad (9)$$

로 주어지고, $W^K_{[i]}$는 M_K에서 $G_{[i]}$의 각 job의 대기시간의 합계를 의미한다.

σ'에 속하는 임의의 그룹 i를 σ의 마지막 위치에 첨가했을때 이를 σ_i라 표시하고, 이때 $J_{[i,j]}$의 가공완료시간은

$$C_{[i,j]}=q_\sigma+S^K_{[i]}+\sum_{m=1}^{j}(w^K_{[i,m]}+P^K_{[i,m]}) \quad (10)$$
$$(j=1, 2, \cdots\cdots,\ n_i)$$

로 나타낼 수 있다.

A_i를 G_i에 속하는 n_i개의 job가운데 이미 순서가 부여된 job을 순서대로 배치한 집합을 나타내고, B_i를 G_i에서 아직 순서가 주어지지 않은 job의 집합을 나타낸다고 한다. $[A_i]+[B_i]=n_i$이다.

임의의 주어진 부분적 그룹순서 σ_i와 G_i에서 주어진 부분적 job 순서 A_i에 대한 총처리시간(makespan)의 하계(lower bound)를 해석하기 위해서 P^k_{ij}를 J_{ij}의 M_k에서 수정가공시간(Revised processing time)을 나타낸다고 하고 다음과 같이 정의한다.

$$P^k_{ij}=S^k_i+\sum_{m=1}^{j}(w^k_{im}+P^k_{im}) \quad (11)$$

총처리시간의 下界는 '최종기계를 기준한 下界' (Last Machine-Based Lower Bound, LMLB)와 '전체기계를 기준한 하계' (Full Machine-Based Lower Bound, FMLB)의 두가지가 있으나 FMLB가 LMLB보다 합리적이므로 FMLB를 사용하기로 한다[11].

$LB\,(F^k_{ij})$를 주어진 부분적 그룹순서 σ_i와 G_i내의 부분적 job순서 A_i가 주어진 경우에 M_k에서 J_{ij}의 총처리시간에 대한 하계를 나타낸다면, $1\leq k\leq K-1$에 대해서

$$LB\,(F^k_{ij})=q_\sigma+P^k_{ij}+\sum_{\substack{m\in B_i\\ m\neq j}} p^k_{im}+\sum_{\substack{l\in\sigma'\\ l\neq i}} Q^k_l + \min_{\substack{l\in\sigma'\\ 1\leq h\leq n_l}}\left\{\sum_{k=k+1}^{K} p^k_{lh},\ \sum_{k=k+1}^{K} p^k_{im}\right\}$$

이고, $k=K$에서는

$$LB\,(F^k_{ij})=q_\sigma+P_{ij}+\sum_{\substack{m\in B_i\\ m\neq j}} p^k_{im}+\sum_{\substack{l\in\sigma'\\ l\neq i}} Q^k_l$$

로 주어지며, 여기서 $P_{ij}=p^K_{ij}$이다. 식(13)은 J_{ij}의 총처리시간의 $LMLB$를 표시한다.

따라서 J_{ij}의 총처리시간에 대한 $FMLB$ 즉 $LB\,(F_{ij})$는

$$LB\,(F_{ij})=\max_{1\leq k\leq K}\{LB\,(F^k_{ij})\} \quad (14)$$

로 정의된다.

또 $LB\,(F^k_i)$를 부분적 그룹순서 σ_i가 주어진 경우에 M_k에서 G_i의 총처리시간에 대한 下界를 나타 낸다면 앞에서와 같은 방법으로, $1\leq k\leq K$에 대해서

$$LB\,(F^k_i)=q^k_{\sigma i}+\sum_{\substack{l\in\sigma'\\ l\neq i}} Q^k_l+\min_{\substack{l\in\sigma'\\ 1\leq h\leq n_l}}\{\sum_{k=k+1}^{K} p^k_{lh}\} \quad (15)$$

이고, $k=K$에 대해서는

$$LB\,(F^K_i)=q_{\sigma i}+\sum_{\substack{l\in\sigma'\\ l\neq i}} Q^K_l \quad (16)$$

이다. 식 (16)은 G_i의 총처리시간에 대한 $LMLB$를 나타낸다.

따라서 G_i 의 *FMLB* 즉 $LB(F_i)$는

$$LB(F_i) = \max_{1 \le k \le K} \{LB(F_i^k)\}$$

로 정의된다.

각 그룹에서 job 순서를 결정하기 위해서 J_{ij} 의 순수납기지연의 상대적 평가척도 α_{ij} 를 다음과 같이 정의하여 도입한다.

$$\alpha_{ij} = \begin{cases} [LB(F_{ij}) - d_{ij}] / [LB(F_{ij} - q_\sigma - P_{ij}], \\ d_{ij} < LB(F_{ij}) \\ 0, \ d_{ij} \ge LB(F_{ij}) \end{cases}$$

만약 $\alpha_{ij} < 1$ 이면 J_{ij} 는 납기보다 늦어지지 않으며, $d_{ij} = 1$ 이면 J_{ij} 의 가공완료시간은 납기 d_{ij} 와 같으며, $\alpha_{ij} > 1$ 이면 J_{ij} 의 가공완료시간이 이미 납기보다 늦어진 경우를 각각 나타낸다. 따라서 α_{ij} 의 값은 전체의 순수납기지연에 대한 J_{ij} 가 미치는 정도를 나타 낸다.

주어진 문제에서 그룹의 순서를 결정하기위해서 G_i 의 순수납기지연의 상대적 평가척도 α_i 를 다음과 같이 정의하여 도입한다.

$$\alpha_i = \frac{1}{n} \sum_{j=1}^{n_i} \alpha'_{ij} \tag{19}$$

단, $n = n_i$ − (G_i 에서 납기지연이 안되는 job의수)

$$\alpha'_{ij} = \begin{cases} [LB(F_i) - d_{ij}] / [LB(F_i) - C_{ij}], \\ \qquad d_{ij} < LB(F_i) \\ 0, \ d_{ij} \ge LB(F_i) \end{cases} \tag{20}$$

만약 G_i 에 포함된 모든 job 이 납기지연이 안되는 경우에 $\alpha_i = 0$ 으로 둔다. 식 (19) 에서 α'_{ij} 는 각 그룹내에서 결정된 job 의 순서가 순수납기지연에 끼치는 영향을 고려하기 위해서 식 (20) 에 의해 다시 계산하였다.

3.2 発見的 技法및 數値計算

앞에서의 예비고찰 및 도입한 개념을 기초로 하여 다음의 발견적 기법을 제안한다[10].

단계 1 : $\sigma = \phi$, $\sigma' = \{1, 2, \cdots\cdots, M\}$ 로 둔다.

단계 2 : σ' 에 속하는 각 그룹에 대한 job순서의 결정단계

(i) $A_i = \phi$, $B_i = \{1, 2, \cdots\cdots, n_i\}$ 로 둔다.

(ii) $|B_i| > 1$ 이면 (iii)으로 가고, 그렇지 않으면 나머지 job 을 n_i 번째에 위치시키고 단계 3 으로 간다.

(iii) B_i 에 속하는 모든 job 에 대하여 α_{ij} 를 계산한다.

(iv) $\alpha^*_{ij} = \max_{j \in B_i} \alpha_{ij}$ 를 주는 job j^*를 찾는다. 만약 $\alpha_{ij}{}^*$의 값을 주는 job 이 여러개인 경우에는 최소납기를 가지는 job 을 택한다.

(v) job j^*를 $(|A_i| + 1)$번째 위치시키고, 이 job 을 B_i 에서 제외한 다음에 (ii)로 가서 반복한다.

단계 3 : $|\sigma'| > 1$ 이면 단계 4 를 가고, 그렇지 않으면 나머지 그룹을 맨마지막에 위치시키고 단계 7 로 간다.

단계 4 : σ' 에 속하는 각 그룹에 대하여 단계 2 에서 결정된 job순서에 따라 α'_{ij} 을 계산하여 α_i 의 값을 계산한다.

단계 5 : $\alpha^*_i = \max_{i \in \sigma'} \alpha_i$ 를 주는 그룹 i^*를 찾는다. 만약 α^*_i 의 값을 주는 그룹이 여러개인 경우에는 그룹내의 job 수가 많은 그룹을 택한다.

단계 6 : 그룹 i^*를 $(|\sigma| + 1)$번째 그룹순서에 위치시키고 이 그룹을 σ' 에서 제외한 다음에 단계 2 로 간다.

단계 7 : 순수납기지연의 합계를 계산하고 과정을 정지한다.

이상의 방법을 예시하기 위해서 다음 표 3 에 주어진 자료를 사용하여 문제를 푼다[10].

Table 3. Production Data for Three-group, Seven-job, Three-stage Group Scheduling Problem

Group		G_1			G_2				G_3		
Job		−	J_{11}	J_{12}	−	J_{21}	J_2	J_{23}	−	J_{31}	J_{32}
Setup / Processing Time		S^k_1	P^k_{11}	P^k_{12}	S^k_2	P^k_{21}	P^k_{22}	P^k_{23}	S^k_3	P^k_{31}	P^k_{32}
STAGE	M_1	6	2	4	3	2	6	3	5	9	4
	M_2	2	4	4	5	3	6	4	7	7	2
	M_3	7	1	6	6	4	5	6	2	3	8
Due Date		−	47	32	−	25	20	32	−	39	60

이 문제에 대한 계산절차는 다음과 같다.

[계산과정 1]

단계 1 : $\sigma = \phi$, $\sigma' = \{1, 2, 3\}$

단계 2 : (a) 그룹 1 의 job순서결정

No.	A_1	B_1	α_{11}	α_{12}	j^*	Job Sequence
1	ϕ	{1, 2}	0. 15000	0. 65714	2	
2	{2}	{1}	−	−	1	$J_{12} - J_{11}$

(b) 그룹 2 의 job순서결정

No.	A_2	B_2	α_{21}	α_{22}	α_{23}	j^*	Job Sequence
1	ϕ	{1,2,3}	0.65789	1.0	0.55556	2	
2	{2}	{1, 3}	0.96970	–	0.80645	1	
3	{2, 1}	{3}	–	–	–	3	$J_{22}-J_{21}-J_{23}$

(c) 그룹 3 의 job 순서결정

No.	A_3	B_3	α_{31}	α_{32}	j^*	Job Sequence
1	ϕ	{1, 2}	0.65116	0.0	1	
2	{1}	{2}	–	–	2	$J_{31}-J_{32}$

단계 3 : $|\sigma'|=3>1$ 이므로 단계 4 로 감.

단계 4 : $\alpha_1=0.4462$, $\alpha_2=0.9652$, $\alpha_3=0.4256$

단계 5 : $i^*=2$

단계 6 : $\sigma=\{2\}$, $\sigma'=\{1, 3\}$, $T_2=0$

〔계산과정 2〕

단계 2 : (a) 그룹 1 의 job 순서

No.	A_1	B_1	α_{11}	α_{12}	j^*	Job Sequence
1	ϕ	{1, 2}	0.52632	1.78571	2	
1	{2}	{1}	–	–	1	$J_{12}-J_{11}$

(b) 그룹 3 의 job 순서 결정

No.	A_3	B_3	α_{31}	α_{32}	j^*	Job Sequence
1	ϕ	{1, 2}	1.0	0.0	1	
2	{1}	{2}	–	–	2	$J_{31}-J_{32}$

단계 3 : $|\sigma'|=2>1$ 이므로 단계 4 로 감.

단계 4 : $\alpha_1=1.2775$, $\alpha_2=0.5357$

단계 5 : $i^*=1$

단계 6 : $\sigma=\{2, 1\}$, $\sigma'=\{3\}$, $T_1=11$.

〔계산과정 3〕

단계 2 : (a) 그룹 3 의 job 순서결정

No.	A_3	B_3	α_{31}	α_{32}	j^*	Job Sequence
1	ϕ	{1, 2}	2.375	0.0	1	
2	{1}	{2}	–	–	2	$J_{31}-J_{32}$

단계 3 : $|\sigma'|=1 \ngtr 1$ 이므로 단계 7 로 감.

단계 7 : $T=T_1+T_2+T_3=11+0+11=22$.

제안된 방법에 의해 얻은 근사최적해는 $T=22$이고, 그룹스케쥴은 $G_2\ (J_{22}-J_{21}-J_{23})-G_1\ (J_{12}-J_{11})-G_3\ (J_{31}-J_{32})$ 이다. 이 예제의 최적해는 발견적 방법에 의해 얻은 그룹스케쥴과 같다.

4. 제안된 方法의 評価

일반적으로 실제적 문제에 대한 解를 간단히 빨리 얻기 위해서 우수한 발견적 기법의 필요성이 증가되고 있으나, 발견적 기법의 유효성을 평가하는 일반적인 방법론이 없다[12, 13]. 그러나 발견적 기법의 유효성을 평가하기 위해서 다음의 방법을 이용할 수 있다.

(1) 최적화기법에 의해 얻은 解와의 비교

(2) 유효성이 인정된 다른 발견적 기법에 의한 解와의 비교

(3) 문헌에 주어진 표준문제에 대한 발견적 기법의 적용

(4) 최적해를 아는 문제를 만들어서 이 문제들에 발견적 기법의 적용

(5) 랜덤 그룹스케듈과의 비교

본 연구에서 제안된 발견적 기법의 유효성을 평가하기 위해서 첫번째 방법은 별로 도움이 안된다. 그 이유는 분지한계법을 도입한 최적화기법[6]이 아주 적은 규모의 문제에만 적용이 가능하고, 또 계산시간이 과다히 소요되기 때문이다. 두번째 및 세번째 방법은 다른 발견적 기법이 제안되어 있지 않고, 문헌에 다루어진 문제가 거의 없기 때문에 사용할 수 없다. 네번째방법은 제한된 범위의 문제에 대해서 적용할 수가 있고, 다섯째 방법은 많은 계산량이 소요되어 실용적이지는 못하지만 가능한 방법이므로 이 두가지 방법을 사용하여 제안된 발견적 기법의 유효성을 고찰하기로 한다.

4·1 最適解를 아는 問題를 사용한 評価

job 의 순수납기지연은 정의에 의해 0 보다 크거나 같으므로, 순수납기지연의 합계가 0 이되는 解는 주어진 문제에 대한 最適解가 된다. 각 job의 납기를 순수납기지연이 0 이 되도록 선정하면 최적해를 아는 문제를 만들 수 있다. 그런데 납가를 아주 여유있게 주면 최적해는 쉽게 얻어지나 이것은 바람직하지 못하므로 가능한한 여유가 없도록 납기를 할당하는 것이 바람직하다. 이의 한가지 방법으로 각 job의 가공완료시간과 같게하는 것으로, 이경우에 각 job의 납기는 각 job에 대한 자료와 그 순서에 의존한다[14].

최적해를 아는 그룹스케듈링문제를 만들기 위해서 각 기계에서 job 의 가공시간 및 그룹준비시간을 랜덤하게 부여하고, 랜덤하게 임의로 선정된 그룹순서와 각 그룹내의 job순서를 사용하여 각 job 의 가공완료시간을 계산하여 이 값을 job 의 납기로 할당한다. 이

같은 절차에 의해서 다음의 조건을 사용하여 96문제를 랜덤하게 만들었다.

(1) 그룹의 수 : 10, 20, 30

(2) 전체job의 수 : 100, 200, 300

(3) 기계의 수 : 8, 12, 16, 20

각 그룹에 속하는 job의 최소개수는 10으로 하였고 (20그룹과 100job, 30그룹과 100job 및 200job의 조합은 제외), 각 그룹에 속하는 job의 수는 같지 않도록 하였다(단 10그룹과 100job, 20그룹과 200job, 30그룹과 300job의 조합은 예외). 각 조합에 대하여 4 문제셋을 랜덤하게 만들었다. job의 가공시간과 그룹준비시간은 1.0과 10.0사이의 구형분포(uniform distribution)을 사용하여 랜덤하게 생성하였다.

시험의 결과는 발견적기법에 의해 얻은 解는 문제의 크기에 관계없이 최적해와 일치하였는데, 이것은 문제를 만드는 과정에서 각 job의 납기의 부여방법과 발견적 기법의 특성때문에 기인한다.

이 결과를 분석하기 위해 납기의 부여방법과 발견적기법을 고찰한다. 만들어진 문제에서 $d_{ij}=C_{ij}$ ($i=1, 2, \cdots, M$; $j=1, 2, \cdots, n_i$)이므로 job순서결정에 사용된 α_{ij}의 값은 모든 i와 j에 대해서 항상 1보다 작거나 같다. 따라서 α'_{ij}의 값 및 α_i의 값도 항상 1보다 작거나 같다. 이 사실은 납기가 지연되는 job이 없으며 따라서 순수납기지연의 합계는 0이다.

위 결과에 의해서 제안된 발견적기법의 유효성에 대한 일반적 결론을 내릴 수 없다. 그 이유는 최적해를 아는 문제를 만들때 사용한 납기의 할당방법에 기인한다. 그러나 앞 결과는 최적해를 아는 특정한 문제에 대해서 발견적방법이 좋다는 결론을 내릴 수 있다.

4·2 랜덤그룹스케듈을 이용한 평가

스케듈링문제는 어렵고 복잡한 조합의 문제이므로 최적해를 열거법에 의해 얻는다는 것은 이론적으로 가능하나 실제적으로는 불가능하다. 하나의 대안으로서 랜덤그룹스케듈에 의해 얻은 解와 발견적기법에 의해 얻은 解를 비교하기로 한다.

시험을 위해 만든 문제는 앞절 (4.1)에 주어진 조건을 사용하였고, 납기는 각 job이 첫번째 기계에서 첫번째 job이라는 가정하에 계산한 완료시간과 마지막 기계에서의 모든 그룹의 그룹가공시간의 합과의 사이의 구형분포에 의해 랜덤한 값을 할당하였다. 시험한 문제는 120문제로써 $VAX-\mathrm{II}$ 컴퓨터를 사용하여 각 문제당 2000개의 랜덤그룹스케듈을 만들어 평가하였다.

표4는 발견적기법에 의해 얻은 그룹스케듈이 각 문제당 2000개의 랜덤그룹스케듈보다 우수한 발생빈도를 나타낸다.

표4에서 보듯이 시험한 120문제중 55%의 문제가 100%의 발생빈도로 랜덤그룹스케듈보다 우수한 근사최적해를 얻을 수 있었고, 평균해서 98%의 발생빈도로써 근사최적해가 랜덤그룹스케듈보다 우수하였다.

표5는 발견적기법에 의한 解와 문제당 2000개의 랜덤그룹스케듈중 가장 좋은 랜덤그룹스케듈보다 우수한 경우를 나타낸다. 시험한 문제의 92%정도가 각 문제당 2000개의 랜덤그룹스케듈중 가장 좋은 解보다 우수함을 보였다. 표5에서 0%의 편차는 발견적기법에 의해 얻은 解가 각 문제당 가장 좋은 랜덤스케듈보다 우수한 경우를 나타낸다. 이 시험과정에서 발견적기법의 유효성과 문제의 매개변수(즉 그룹의 수, 그룹내의 job수, 기계의 수)사이에는 아무런 관계가 없음이

Table 4. Comparison of Number of Times Between Heuristic Solutions and Random Solutions

Number of Times (%)	Number of Problems	Percentage of Problems (%)	Cumulative Percentage of Problems (%)
100	66	55.00	55.00
99.0 − 99.99	32	26.68	81.68
98.0 − 98.99	7	5.84	87.52
97.0 − 97.99	4	3.33	90.85
96.0 − 96.99	4	3.33	94.18
95.0 − 95.99	1	0.83	95.01
94.0 − 94.99	1	0.83	95.84
93.0 − 93.99	0	0.00	95.84
92.0 − 92.99	0	0.00	95.84
91.0 − 91.99	1	0.83	96.67
90.0 − 90.99	0	0.00	96.67
80.0 − 89.99	4	3.33	100.00
Total	120	100.00	

발견되었다.

이상의 2가지 접근법에 의한 제안된 발견적 기법의 평가결과는 시험에 사용한 문제에 대하여 좋은결과를 주는 것이 밝혀졌다.

Table 5. Deviation of Heuristic Solution from the Best Random Solution

Deviation (%)	Percentage of Problems (%)	Cumulative Percentage of Problem (%)
0	55.00	55.00
0.001-0.999	9.17	64.17
1.00-1.999	6.66	70.83
2.00-2.99	8.33	79.16
3.00-3.99	9.17	88.33
4.00-4.99	3.33	91.66
5.00-5.99	1.67	93.33
6.00-6.99	1.67	95.00
7.00-7.99	0.00	95.00
8.00-8.99	1.67	96.67
9.00-10.0	3.33	100.00

4·3 所要計算時間

문제의 크기와 계산시간사이의 관계 및 제안된 발견적기법이 짧은 계산시간으로 큰 크기의 실제적 문제를 풀 수 있는가를 고찰하기 위해서 여러개의 큰 규모의 문제를 사용하여 계산시간을 분석하였다.

표6은 125가지의 상이한 문제를 만들어서 소요계산시간을 측정한 결과이다. 이 시간에는 문제를 만드는 시간은 제외되었고, 결과는 *VAX*-Ⅱ 컴퓨터를 사용하여 얻었다. 표6에 주어진 시간은 각 문제사이즈당 25개의 문제를 만들어 계산한 시간의 평균값이다.

Table 6. Problem Size vs. Computation Time

Size of Problems			Time (sec)
Number of Groups	Total Number of Jobs	Number of Stages	
10	100	10	10.38
15	150	10	27.91
20	200	10	57.22
25	250	10	102.45
30	300	10	157.53

계산에 소요된 시간은 문제의 크기에 따른 指數的增加를 보여주지 않는다. 문제의 크기가 증가함에 따라 계산시간이 증가하는데 이것이 그룹의 수에 기인하는 것인지 job의 수에 기인하는지를 고찰한 결과 그룹의 수보다 job의 수에 더 의존하였으며, 또 일정한 그룹의 수와 전체 job의 수가 주어진 경우에 기계대수의 변화는 계산의 소요시간에 큰 영향을 미치지 않음을 알 수 있었다.

제안된 발견적기법은 비교적 짧은 시간만에 좋은 解를 쉽게 얻을 수 있으며, 소요계산시간은 그룹의 수보다 job의 수에 의존함이 밝혀졌다.

5. 結 論

(1) 그룹테크노로지의 개념을 도입한 그룹스케쥴링 모델을 다품종, 다단계 생산시스템에서 납기제약조건하에 설정하여 분석 및 고찰하였다.

(2) 순수납기지연의 합계를 최소화하는 그룹순서와 각 그룹내의 job순서를 결정하는 발견적기법을 개발하여 제안하였고, 예제를 사용하여 제안된 기법을 예시하였다.

(3) 제안된 발견적기법의 유효성을 최적해를 아는 문제나 랜덤그룹스케듈에 적용하여 그 유효성을 분석한 결과로 빠른 시간내에 비교적 좋은 解를 얻을 수 있음을 알 수 있었다.

〈감사의 글〉

본 연구는 1982년도 문교부의 學術硏究助成費의 지원을 받아 수행된 것으로서 이에 감사드립니다. 특히 본 연구를 위해 조언과 격려를 해 주신 미국펜실바니아주립대학교 공과대학 산업공학과의 咸仁英敎授님께 충심으로 감사를 드리는 바입니다.

参考文献

1. Ham, I., "Introduction of Group Technology," SME Technical Paper MMR-76-03, February 1976; also CAM-I Seminar Proceedings, June 1975 (No. P-75-ppp-01) and January 1976 (No. P-76-ppp-01).
2. Hitomi, K., and Ham, I., "Group Schedulıng Technigues for Multi-Product, Multi-Stage Manufacturing Systems," ASME Papper #77-WA/Prod-27, August 1977, pp. 759-765.
3. Yoshida, T., Nakamura, N., and Hitomi, K., "A Study of Group Scheduling," Transactions of Japan Industrial Management Association, Vol. 28, No. 3, 1977, pp. 323-328 (in Japanese).
4. Nakamura, N., Yoshida, T., and Hitomi, K., "Group Prodution Scheduling for Minimum Total tardiness Part (I)," AIIE Transactions, Vol. 10, No. 2, 1978, pp. 157-162.
5. Hitomi, K., and Ham, I., "Operations Scheduling for Group Technology Applications," CIRP Annals, Vol. 25, Hallwag Ltd., Bern, Switzerland, August 1976, pp. 419-422.
6. Nakamura, N., and Hitomi, K., "Optimization of Group Scheduling for the Multiple Production Stages," Transactions of Japan Society of Mechanical Engineers, Vol. 42, No. 361, 1976, p. 2964-2973 (in Japanese).
7. Ham, I., and Hitomi, K., Nakamura, N., and Yos Yoshida, T., "Optimal Group Scheduling and Machine-Speed Decision under Due-Date Constraints," ASME Papper No. 78-WA/Prod-39, 1979, pp. 1-7.
8. Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, New York, 1974,
9. King, J. R., and Spachis, A. S., "Heuristics for Flow-Shop Scheduling," International Journal of Production Research, Vol. 18, No. 3, 1980, pp. 345-357.
10. Cho, K., Group Scheduling Under Due Date Con Constraints in Multi-Stage Manufacturing Systems, Ph. D. Thesis, The Pennsylvania State University, November 1982.
11. Baker, K. R., "A Comparative Study of Flow-Shop Algorithms, Operations Research,, Vol. 23, No. 1, January-February 1975 pp. 62-73.
12. Ignizio. J. P., "Solving Large-Scale Problems: A Venture into a New Dimension," Journal of Operational Research Society, Vol. 31, 1980, pp. 217-225.
13. Zanakis, S. H., and Evans, J. R., "Heuristic Optimization: Why, When, and How to Use It," Interfaces, Vol. 11, No. 5, October 1981, pp. 84-90.
14. Baker, K. R., and Bertrand, J. W. M., "A Comparison of Due-Date Selection Rules," AIIE Transactions, Vol. 13, No. 2, June 1981, pp. 123-131,