

FMS운용을 위한 일정계획용 소프트웨어
—Development of Scheduling Software
for Flexible Manufacturing System—

윤 덕 균 *
황 의 철 **

Abstract

This paper is concerned with software developments for scheduling and sequencing of FMS. The scheduling algorithms are developed for 3 types of FMS:single machine type FMS, flowshop type FMS, assembly line type FMS. For the single machine type FMS, full enumeration algorithm is used. For the flowshop type FMS heuristic algorithms are developed. For the assembly type FMS the existing PERT/CPM algorithm is applied. Numerical examples are presented for illustration of each algorithm. Each software program list are attached as appendices.

1. 서 언

일정계획에 관한 연구는 1년에 약 100여편의 결과가 발표되고 있으나 완벽한 FMS용 일정계획 기법을 주제로 한 논문은 거의 없다 하겠다. 업계한 의미에서 FMS는 흐름작업 지향적인 Jobsop형 생산시스템이므로 그 일정계획은 복잡한 양태를 뛸 수밖에 없다 하겠다, 이중에서도 FMS 시스템에 가장 가까운 기법의 하나로 GT(group technology)기법을 이용한 일정계획기법들이 나타나고 있다. Hittomi[22]~[24] 등은 GT를 이용하여 그를 일정계획하는 기법을 최근에 발표하고 있으며, Yosida[50] 등도 비슷한 기법을 제시하고 있다.

그러나 FMS에 직접 쓸 수 있는 기법은 아니며 새로운 FMS의 시스템 분석에 의하여 적합화된 새로운 기법을 절실히 필요로 한다 하겠다. 이러한 시도는 지금까지 기개발된 흐름작업 일정계획기법[1], [8], [21], [26], [33]에 jobsop의 일정계획[5]~[7], [19], [35], [45]을 접목시키는 작업이 될 것이다.

이러한 시도로서 일본의 大橋等[52], 岩田[53] 등은 FMS시스템의 일정계획 기법에 대한 새로운 시도를 보여주고 있다.

그러나 이러한 FMS의 일정계획 작성 용 알고리즘의 개발은 FMS의 다양성 때문에 어려움을 갖게 된다. 이에 본 연구에서는 FMS를 다음과 같이 3종류로 분류 각각에 대해서 알고리즘과 소프트웨어를 개발하기로 한다.

- (1) 단일설비형 FMS
- (2) 흐름작업형 FMS
- (3) 조립작업형 FMS

2. 단일 설비시스템의 일정계획

그림 1에서 예시하는 바와 같이 단일설비 시스템은 M/C(machining center)자동공작물 공급장치, 패랫풀장치로 구성되어 있다. 일종의 FMC(flexible manufacturing cell)이라고 볼 수 있는 부설비는 광의의 FMS의 범주에 든다 하겠다.

이 논문은 1988년 문교부 지원 한국학술진흥재단의 자유·광보과제 학술연구조성비에 의하여 연구되었음.

*한양대 산업공학과 교수

접수: 1991. 5. 30.

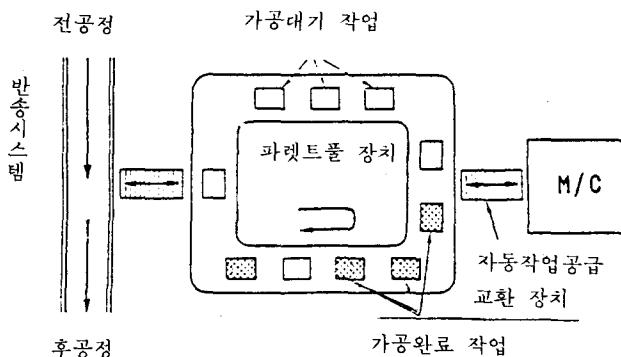


그림 1. 단일설비 시스템의 예

단일 설비에서의 일정계획 문제는 여러개의 작업을 어떤 순서로 작업할 것인가하는 작업순서계획이다. 이러한 순서계획의 작성에는 다음과 같은 기본적인 가정을 필요로 한다.

- (1) n 개의 독립된 작업이 있으며 각 작업은 순서적으로 단일설비에서 수행되어야 하며 동시에 두 개 이상의 작업을 할 수 없다.
- (2) 각 작업의 준비시간(Set up time)은 순서와 무관하게 각 작업에 따라 결정하며, 이를 작업시간에 포함시켜 생각해도 무방하다.
- (3) 작업의 납기, 작업시간 등은 미리 알려져 있다.
- (4) 단일설비는 계속해서 가동이 가능하며 작업을 기다리는 한 작업을 계속해야 한다.
- (5) 한번 시작된 작업은 중도에서 끝낼 수 없으며 완성되어야 한다.
- (6) n 개의 작업은 모든 시간 0에서 작업이 가능하여야 한다.

이러한 조건하에서 n 개의 작업의 처리순서와 작업번호의 순열(Permutation) 사이에는 일대일의 대응관계가 성립되어 총 $n!$ 개의 서로 다른 순서가 존재하게 된다. 여러 개의 설비가 존재하는 경우 각각의 기계에 대해서 $n!$ 개의 순서가 존재하게 되어 m 개의 기계라면 $(n!)^m$ 의 일정계획이 존재하게 되나 이를 간편히 하고자 하여 첫번째 기계에서 정해진 작업순서를 그 다음 기계에서도 동일하게 유지하는 경우 특별히 이를 순열 작업순서(Permutation Schedule)라 한다.

순열일정계획을 기술하는 방법으로 []를 사용하는데 예를 들면 [5]=2라는 의미는 5번째 처리되는 작업번호가 2임을 나타내니다.

단일설비모델에서 주어지는 작업의 속성은 다음과 같은 정보로 주어진다.

- 작업시간(t_i) : 작업 j 를 처리하는데 요구되는 작업시간
 준비시간(r_j) : 작업을 위해서 작업 j 의 준비가 완료된 시간
 납기예정시간(d_i) : 작업 j 의 작업이 완료되도록 예정된 납기시간

여기서 작업시간은 일반적으로 가동준비를 위한 시간을 포함하며, 준비시간은 작업의 처리를 위해서 작업설비에 도달한 시간을 의미한다. 그러나 일반적인 문제에서는 가정 (6)에 의해서 모든 작업에서 $r_j=0$ 이 된다. 납기예정일은 문제에 따라서는 특별히 제시되지 않는 경우도 있으나 일정계획의 본질상 작업의 데드라인을 맞춘다는 의미는 일정계획기의 기본적인 관심사이다.

주어진 작업의 속성을 이용하여 작업순위를 작성한 결과 나오는 모든 정보는 그 작업순서의 경제성을 평가하는 지표를 제공하게 된다. 그러한 자료는 다음과 같은 것을 포함하게 된다.

- (1) 작업완료시간(Complete Time : C_j) : 작업 j 가 완료된 시간
- (2) 작업흐름시간(Flow Time : F_j) : 작업 j 가 시스템에서 소비한 시간, 즉 완료된 시간에서 도착한 시간을 뺀 $C_j - r_j$ 가 된다.
- (3) 지연시간(Lateness : L_i) : 작업완료시간이 납기예정일을 초과한 시간, 즉 $L_i = C_j - d_i$

여기에서 작업흐름시간 F_j 는 각각 작업 j 에 대한 시스템의 반응의 신속성에 대한 척도이며, 작업이 도착해서 출발할 때까지의 시스템 내에서 머무는 기간을 의미한다.

지연시간 L_j 는 주어진 납기에 대한 시스템의 신뢰성에 대한 척도로서 지연시간 L_j 가 음수가 되면 작업을 조기 달성하게 되는 것으로 이것은 요구하는 서비스보다 과잉 서비스를 제공하게 되어 이것 또한 재고유지비 등을 부담하는 결과가 초래된다. 특별히 양수의 지연시간 만이 지체시간(tardiness)으로서 관심의 대상이 되는 경우가 있다. 이를 다음과 같이 정의한다.

납기지체시간(tardiness T_j) : 작업 j 의 지연이 양수인 경우로 납기를 맞추지 못해 지체상금을 물어야 하는 경우로서 $T_j = \max |0, L_j|$ 가 된다.

주어진 일정계획은 일반적으로 작업의 완료시간으로 주어지는 총괄적인 척도에 의하여 평가되며 결정되게 되는데 총괄적인 척도는 일반적으로 다음과 같은 척도를 포함한다.

$$(1) \text{ 평균작업흐름시간} : \bar{F} = \frac{1}{n} \sum_{j=1}^n F_j$$

$$(2) \text{ 평균작업지체시간} : \bar{T} = \frac{1}{n} \sum_{j=1}^n T_j$$

$$(3) \text{ 최고작업흐름시간} : F_{\max} = \max |F_j|$$

$$(4) \text{ 최고작업지체시간} : T_{\max} = \max |T_j|$$

$$(5) \text{ 지체작업수} : NT = \sum_{j=1}^n (T_j)$$

$$\text{단, } (X) = \begin{cases} 1 & \text{만약 } X > 0 \\ 0 & \text{그외} \end{cases}$$

이러한 척도는 작업완료시간의 합수가 되므로 이를 수식으로 표시하면

$$Z = f(C_1, C_2, \dots, C_n)$$

이 된다. 그런데 이러한 일정계획의 척도는 다음과 같은 성질을 갖고 있게 되는데 이를 특별히 정상적수행척도(regular measure of performance)라 부른다.

(1) 일정계획의 목적이 Z 를 최소화하는 것이다.

(2) Z 는 작업완료시간 C_j 의 하나라도 증가하게 되면 Z 도 증가한다.

일정계획의 평가척도로서 평균작업흐름시간이라든지 평균지체시간 등 단순척도에 대한 일정계획기법에 대해서는 이미 언급한 바와 같다. 그러나 이것이 복합화된 다중목표하에서는 일반법칙으로 단순화하기는 어려우며 이에 대한 해법 또한 다양하게 제시되어 있다.

평균작업대기시간도 줄이고 싶고 작업지체시간도 줄이고 싶은 다중목표를 하나의 공통척도, 즉 비용으로 환산해서 최소화하는 작업이 수행된다. 이 경우 대기시간에는 단위 대기비용의 가중치를 부과하고 지체시간에는 단위 지체비용의 가중치를 곱해서 그 합을 목적함수로 해서 최소화하는 방법을 택하는 것이다.

이러한 과정에서 고려하는 비용에는 다음과 같은 것이 포함된다.

(1) 대기비용 : 작업이 처리되기 전에 대기하는 시간 중 소요되는 비용

(2) 재고비용 : 작업이 납기보다 먼저 끝남으로써 재고가 발생되어 소요되는 비용

(3) 지연간접비용 : 지연일수에 상관없이 지체되는 경우 지불되어야 하는 고정비 성격의 지체상금

(4) 지체비용 : 지연일수에 비례해서 지불되는 지체상금

(5) 특별지체비용 : 정해진 일수에 작업이 완료되지 못하면 특별히 추가로 지불해야 되는 비용

이러한 비용을 최소화하는 최적해를 구하는 완전한 알고리즘은 아직까지 개발되지 못하고 있다. 다만 이에 대한 근사해법이 제시되어 있는 바, 이를 열거하면 다음과 같다.

(1) 인접작업상호교환법(Adjacent Pairwise Interchange Method) : 먼저 EDD 또는 SPT 순위에 의하여 작업순위를 결정한 후에 인접작업을 상호교환해 가면서 목적함수 값을 줄여가는 방법.

- (2) 動的計劃法(Dynamic Programming Approach) : 정상수행척도 Z 가 덧셈형으로 되어 있는 경우, 동적 계획법을 사용하여 목적함수값을 최소화 하는 방법.
- (3) 분기한계법(Branch & Bound Approach)
- (4) 인근탐색법(Neighborhood Search Method)
- (5) 무작위샘플링법
- (6) 全數計算法

위의 방법 중 최적해를 제시해 주는 방법은 전수계산법으로 가장 비효율적인 것 같으면서도 어느 의미에서는 오늘날과 같이 컴퓨터가 발달된 시점에서는 시간이 조금 걸리지만 가장 확실하게 최적해를 보장하는 방법이 됬다. 컴퓨터의 속도가 빨라지면 빨라질수록 이러한 경향은 더욱 심화될 것으로 생각되며 본 프로그램은 이러한 바탕 위에 모든 순열작업순위에 대해서 목적함수를 평가해서 최적의 해를 찾는 기법을 사용하고 있다.

표 1. 컴퓨터 수행예제

작업번호	작업시간	납기일	대기비용	재고비용	지연간접비용	지체비용
1	5	8	2	5	50	10
2	11	15	3	6	40	15
3	2	10	2	4	50	5
4	9	18	4	4	55	8

표 1과 같이 주어진 4개의 작업에 대해서 본 프로그램을 이용해서 작업순위를 결정하여 보자. 먼저 프로그램을 수행시키면 다음과 같이 프로그램의 시작을 알리는 메세지가 나온다.

단일형 FMS의 작업순위결정용 소프트웨어

한양대학교 산업공학과
공장자동화 시스템 연구실
전화 292-2111 (교)2306
교수 윤덕균

데이터 입력

작업의 수는 얼마입니까? 4

작업번호, 작업일수, 납기일, 대기비용, 재고비용, 지연간접비용, 지체비용을 입력하시요

? 1, 5, 8, 2, 5, 50, 10

? 2, 11, 15, 3, 6, 40, 15

? 3, 2, 10, 2, 4, 50, 5

? 4, 9, 18, 4, 4, 55, 8

위와 같이 작업일수, 납기일, 대기비용, 재고비용, 지연간접비용, 지체비용을 입력하고 나면 다음과 같이 입력자료에 대한 확인문의가 나오게 된다.

입력데이터 확인

1	5	8	2	5
50	10			

2	11	15	3	6
40	15			
3	2	10	2	4
50	5			
4	9	18	4	4
55	8			

이와 같은 절차로 4개의 작업에 대한 정보입력이 끝나고 나면 작업지연이 어떤 특정일수를 넘는 경우 물어야 되는 특별지체상금에 대한 입력을 묻게 된다. 없는 경우는 "N"을 입력시키면 되고, 있는 경우에는 "Y"를 입력시킨 후에 지체상금을 특별히 지불해야 하는 특정일수와 추가특별지체상금을 입력하면 된다.

*작업의 지연이 어떤 특정일수를 넘는 경우 추가비용을 부과 하시겠습니까 (Y/N) Y

*특정일수를 얼마로 하시겠습니까 20

*추가비용은 얼마입니까 16

이것으로 모든 입력작업은 완료되게 되며 곧이어 계산결과가 출력된다. 세일 먼저 최소여유시간 우선순위법에 의한 작업순위나 그 때의 완성일, 대기일수, 지연일수, 초기완료일수, 그리고 작업비용이 다음과 같이 출력된다.

최소여유시간 우선

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
1	5	8	5	0	0	3
2	11	15	16	5	1	0
3	2	10	18	16	8	0
4	9	18	27	18	9	0
합계	27	51	66	39	18	3
평균				9.75	4.5	.75

작업순서 : 1 2 3 4

위의 작업순서에 의한 비용은 406

그 다음 납기일 우선순위(EDD)에 의한 작업순위가 출력된다.

납기할 우선순위법

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
1	5	8	5	0	0	3
3	2	10	7	5	0	3
2	11	15	18	7	3	0
4	9	18	27	18	9	0
합계	27	51	57	30	12	6
평균				7.5	3	1.5

작업순서 : 1 3 2 4

위의 작업순서에 의한 비용은 342

곧이어 최소작업시간(SPT) 우선순위에 의한 작업순위 결과가 다음과 같이 비용 345와 함께 출력된다.

최소작업시간 우선순위법

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
3	2	10	2	0	0	8
1	5	8	7	2	0	1
4	9	18	16	7	0	2
2	11	15	27	16	12	0
합계	27	51	52	25	12	11
평균				6.25	3	2.75

곧이어 지연될 작업의 수가 최소인 작업순위에 의한 계산결과가 출력된다.

지연된 작업의 수가 최소인 작업순서

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
1	5	8	5	0	0	3
3	2	10	7	5	0	3
4	9	18	16	7	0	2
2	11	15	27	16	12	0
합계	27	51	55	28	12	8
평균				7	3	2

작업순서 : 1 3 4 2

위의 작업순서에 의한 비용은 341

계속해서 평균지연이 최소인 작업순서에 의한 작업순위결과가 출력된다.

평균지연이 최소인 작업순서

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
3	2	10	2	0	0	8
1	5	8	7	2	0	1
4	9	18	16	7	0	2
2	11	15	27	16	12	0
합계	27	51	52	25	12	11
평균				6.25	3	2.75

작업순서 : 3 1 4 2

위의 작업순서에 의한 비용은 345

마지막으로 최적작업순서가 비용 341과 함께 다음과 같이 출력된다.

최적 작업순서

작업번호	작업일수	납기일	완성일	대기일수	지연일수	초기완료일수
3	2	10	2	0	0	8
1	5	8	7	2	0	1
4	9	18	16	7	0	2
2	11	15	27	16	12	0
합계	27	51	52	25	12	11
평균				6.25	3	2.75

작업순서 : 1 3 4 2

위의 작업순서에 의한 비용은 341

3. Flow Shop에서의 作業順位決定技法

본 프로그램으로 전산화한 작업순위 결정 알고리즘은 Gelders와 Sambandam[1]의 근사해법에 기초를 두고 이를 우리 실정에 맞도록 변형한 것이다.

즉, 본 프로그램은 근사해법으로서 다음과 같은 3가지 우선순위법칙을 사용하여 작업순위를 정한다.

- (1) 動的作業與置의 원칙
- (2) 재고비용(holding cost)이 큰 비용에 대한 우선순위
- (3) 지체시간이 최소화되도록 작업우선순위

이상과 같은 우선순위를 이용하여 작업우선순위를 정해가는 본 프로그램의 논리를 간략하게 전개하도록 해보자, 자세한 알고리즘을 원하는 독자는 참고문헌[1]을 참조바란다.

먼저 논의상 편의를 위해서 기호를 정의한다.

$j =$ 작업번호 ($j=1, 2, \dots, J$)

$m =$ 기계번호 ($m=1, 2, \dots, M$)

$p_{jm} = m$ 기계에서의 j 작업의 가공시간

$d_j = j$ 작업의 성해진 납기

$w_j = j$ 작업의 연체 단위 기간당 연체 상금

$h_j = j$ 작업의 단위 기간당 재고비용

σ : 이비 작업순위가 할당된 작업들의 집합

$\bar{\sigma}$: 아직도 자작업순위가 미할당된 작업의 집합

$t_{jm} = j$ 작업의 m 기계에서의 작업완료시간

본 문제에서의 목적함수는 각 작업의 연체상금액과 재고비용의 합을 최소로 하는 작업순위계획을 작성하는 것이기 때문에 목적식은 다음과 같이 된다.

$$\text{최소화} : Z = \sum Z(t_{jm}) = \sum \{W_j \max(t_{jm} - d_j; 0) + h_j t_{jm}\}$$

여기서 $\max(t_{jm} - d_j; 0)$ 은 작업 j 의 납기자연시간이므로, $w_j \max(t_{jm} - d_j; 0)$ 은 납기연체비용이며 $h_j t_{jm}$ 은 j 작업의 j 의 재고비용이 된다. 목적함수식에서 볼 수 있는 바와 같이 각 작업의 t_{jm} 즉, 최종기계($=M$)의 완료시간을 최소화하는 작업순위의 일정계획을 작성해야 하는 것이다.

$t(\sigma, m)$ 을 σ 작업순위 집합으로 작업순서가 결정되었을 경우의 m 기계에서의 작업완료시간으로 정의하자. 여기에서 j 작업을 σ 작업집합 바로 다음에 할당할 경우, m 기계에서의 j 작업의 완료시간 $t(\sigma_j, m)$ 은

$$t(\sigma_j, m) = \max[t(\sigma, m); t(\sigma_j, m-1)] + p_{jm} \quad (1)$$

이때 j 작업의 m 기계에서의 유휴시간 $\Delta_{j, m}$ 은 다음과 같다.

$$\Delta_{j, m} = \max[t(\sigma_j, m-1) - t(\sigma, m); 0] \quad (2)$$

이러한 유휴시간을 사용하여 각 작업의 마지막 기계의 공정시간을 새로 정의하면

$$p_j = \Delta_{j, M} - p_{jM} \quad (3)$$

S 를 Δ 로, 작업배치한 마지막 기계의 작업완료시간이라 하면 즉,

$$S = t(\sigma, M) \quad (4)$$

이 때 T_j 를 부분작업순위 σ_j 의 총작업완료시간의 하한으로 정의하면 다음과 같이 구할 수 있다.

$$T_j = t(\sigma_{j, M}) + \sum_{m=1}^{j-1} p_{jM} \quad (5)$$

j 작업의 모든 기계에 서의 지체시간의 합 Δ_j 를 구하면 다음과 같다.

$$\Delta_j = \sum_{m=1}^M \Delta_{j,m} = \sum_{m=2}^M \Delta_{j,m} \quad (6)$$

계산의 편의상 $\sum_{m=1}^M \Delta_{j,m} = 0$ 일 때는 $\Delta_j = 1$ 로 놓고 다음과 같이 연체비용의 척도로서 다음과 같이 R_j 를 정의하여 작업순위 결정의 척도로 삼는다.

$$R_j = R_j / \Delta_j \quad (7)$$

$$\text{단, } R_j' = (x_{1j}w_j + x_{2j}h_j) / p_j \quad (8)$$

여기서

$$\begin{aligned} x_{1j} &= (T_j - d_j) / (T_j - S - P_j) \\ \text{만일 } d_j &< T_j \\ 0 \text{ 그 외에,} \end{aligned} \quad (9)$$

$x_{2j}=1$ 이상의 가중치(본 예제에서는 1로 하고 있음)

이를 요약하여 단계별로 써 보면 다음과 같다.

단계 0: $\sigma = \emptyset$, $\sigma = \{1, 2, \dots, M\}$ 으로 놓는다.

단계 1: $|\sigma| > 1$ 이면 2단계로 가고 그렇지 않으면 σ 로부터 차례로 작업을 배치하여 $Z(t_{j,M})$ 을 제거한다.
단계 5로 간다.

단계 2: σ 집합의 모든 작업에 대해서 식(6)~(9)을 이용해서 R_j 를 계산한다.

단계 3: R_j 값을 최대로 하는 j^* 를 찾아낸다.

단계 4: 작업 j^* 를 $|\sigma| + 1$ 순위에 배치하고 σ 로부터 제거한다. 식(1)을 사용해서 $Z(t_{j^*,M})$ 을 계산하고 단계 1로 간다.

단계 5: 총비용 $\sum_{j=1}^J Z(t_{j,M})$ 을 계산한다.

단계 6: $j-1$ 개의 인접 작업을 상호 교환하여 더 좋은 해를 확인한다.

단계 7: 더 좋은 해가 있으면 찾아 보고 그렇지 못하면 끝낸다.

이 절차를 표 2 예제를 가지고 설명해 보도록 하자.

표 2. 예제 입력자료

작업번호	작업시간					남기일	지체비용	재고비용
	1	2	3	4	5			
1	8	10	9	2	1	32	2	3
2	10	1	1	2	8	35	5	4
3	1	2	4	8	1	41	7	3
4	7	5	3	3	7	46	8	1
5	1	10	1	6	4	52	6	5

단계 0: $\sigma = \emptyset$, $\sigma = \{1, 2, 3, 4, 5\}$

반복 1:

단계 1: $|\sigma| = 5$ 으로 단계 2로 간다.

단계 2: $j=1$ 에 대해서 그림 2에서 볼 수 있는 바와 같이

$$\Delta_{1,1}=0, \Delta_{1,2}=8, \Delta_{1,3}=18, \Delta_{1,4}=27, \Delta_{1,5}=29$$

$$P_1 = \Delta_{1,5} - P_{1,5} = 29 + 1 = 30$$

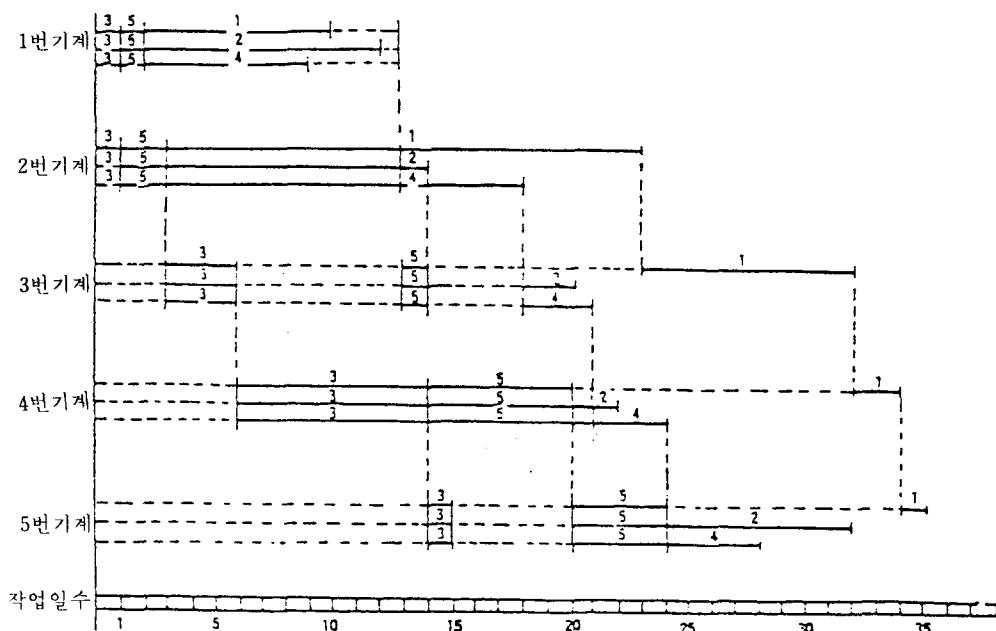


그림 3. 3번 작업과 5번 작업이 1, 2 순위로 배치된 후 3순위로 1, 2, 3 작업을 각각 배치한 경우의 간트차트

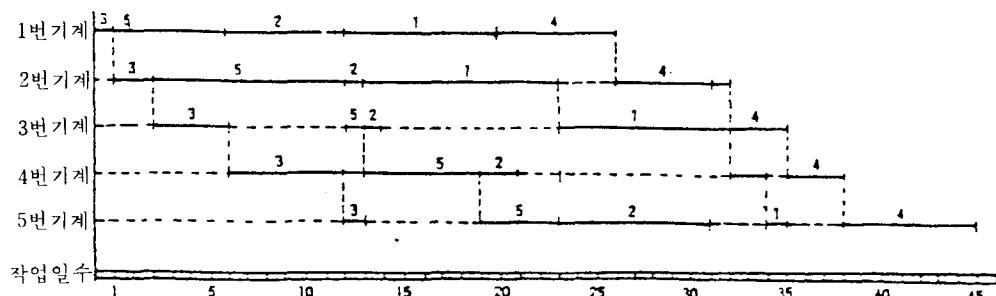


그림 4. 3-5-2-1-4의 순위로 작업을 배치할 경우의 간트차트

표 3. 계산결과요약

반복	단계 1		단계 2					단계 3		단계 4	
	σ	σ	R_1	R_2	R_3	R_4	R_5	$J_1 * \sigma$	$\sigma Z(t_i^*, m)$		
(1)	ϕ	$ 1, 2, 3, 4, 5 $	0.0040	0.0039	0.0072	0.0008	0.0054	3 3	$ 1, 2, 4, 5 $	48	
(2)	$ 3 $	$ 1, 2, 3, 5 $	0.0147	0.0384	—	0.0067	0.0505	5 3, 5	$ 1, 2, 4 $	125	
(3)	$ 3, 5 $	$ 1, 2, 4 $	0.0357	0.9688	—	0.0357	—	2 3, 5, 2	$ 1, 4 $	132	
(4)	$ 3, 5, 2 $	$ 1, 4 $	0.0257	—	—	0.0079	—	1 3, 5, 2, 1	$ 4 $	356	

3. 프로그램 운용사례

표 2에서 볼 수 있는 바와 같이 5대의 기계에서 5개의 작업을 수행하는 주문생산시스템에서의 작업순위 결정을 본 프로그램으로 수행해보도록 하겠다.

먼저 프로그램을 수행시키면 다음과 같이 프로그램의 시작을 알리는 메세지가 나온다.

$$S=0 \quad (\because \sigma = \phi) \\ T_1 = 30 + (8+1+7+4) = 50$$

식 (9)로부터

$$x_{1,1} = (50-32)/(50-0-30) = 18/20 = 0.90$$

$$x_{2,1} = 1$$

$$R_1' = (0.9 \times 2 + 1 \times 8)/30 = 0.3267$$

$$\Delta_1 = (8+18+27+29) = 82$$

$$R_1 = 0.3267/82 = 0.0040$$

같은 방법으로 작업 $j = |2, 3, 4, 5|$ 로부터 R_j 값을 구하면,

$$R_2 = 0.0039, R_3 = 0.0072, R_4 = 0.0008, R_5 = 0.0054$$

단계 3 : R_j 값을 최대로 하는 j^* 는 3이다.

$$\text{단계 } 4 : \sigma = |3|, \sigma = |1, 2, 4, 5|, Z(t_{1,5}) = 48$$

이로써 다시 단계 1로 넘어간다.

이러한 반복과정은 그림 3과 그림 4를 거쳐서 진행된다. 이를 요약한 것이 표 3이다.

반복 5 :

단계 1 : $|\sigma| = 1$ 으로 우리는 마지막에 4작업을 배치한다. $\sigma = |3, 5, 2, 1, 4|, Z(t_{1,5}) = 100$ 을 계산하고 단계 5로 넘어간다.

단계 5 : 작업순위를 $|3, 5, 2, 1, 4|$ 로 정하고 총비용 = $(48+125+132+356+100) = 761$ 을 계산한다.

단계 6 : 인접한 두 작업순위를 바꿔가면서 총 비용을 계산한다.

$$\text{순서} = 5-3-2-1-4 \quad \text{총비용} = 796$$

$$\text{순서} = 3-2-5-1-4 \quad \text{총비용} = 807$$

$$\text{순서} = 3-5-1-2-4 \quad \text{총비용} = 771$$

$$\text{순서} = 3-5-2-4-1 \quad \text{총비용} = 771$$

단계 7 : 더 좋은 해가 없으므로 여기서 중단한다.

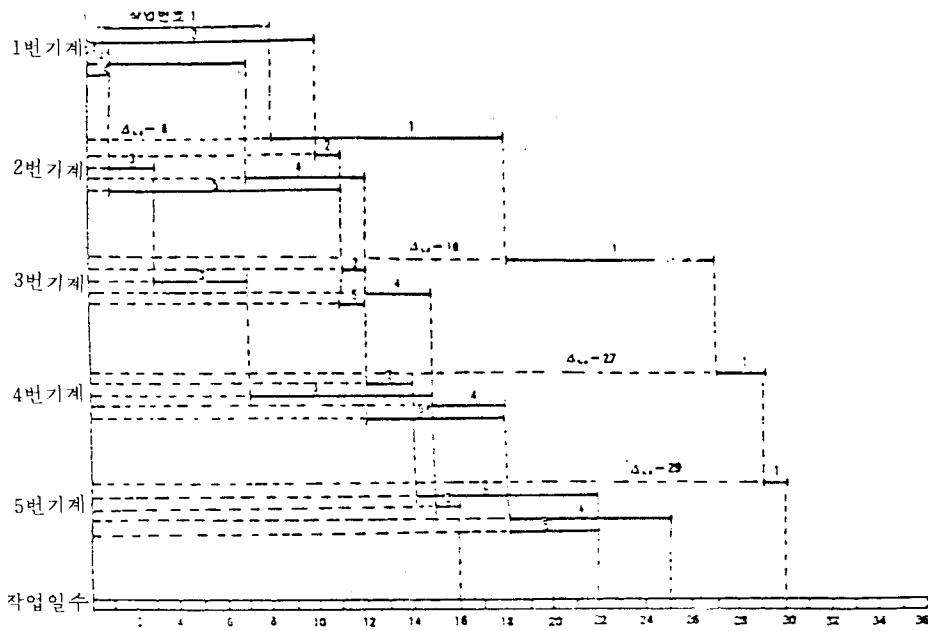


그림 2. 각 작업을 1번 순위로 배치할 경우의 간트차트

FLOW SHOP형 FMS의 일정계획용 소프트웨어

한양대학교 산업공학과
공장자동화 시스템 연구실
전화 292-2111 (교)2306
교수: 윤 덕 균

그 다음 자료입력에 대한 메세지가 나오게 되는데 순서에 따라서 자료를 입력시키면 된다.
먼저 작업의 수와 기계의 수의 입력을 요구한다.

《데이터입력》

작업의 수는 5
기계의 수는 5

그리고 곧이어 기계 각각에 대한 작업들의 작업시간을 입력시키도록 한다. 본 예제에서는 5대의 기계와 5개의 작업이므로 25개의 작업시간을 다음과 같이 입력시키게 된다.

《기계와 작업에 대한 시간입력》

작업 1과 기계 1에 대한 시간 8	작업 2와 기계 5에 대한 시간 8	작업 4와 기계 4에 대한 시간 3
작업 1과 기계 2에 대한 시간 10	작업 3과 기계 1에 대한 시간 1	작업 4와 기계 5에 대한 시간 7
작업 1과 기계 3에 대한 시간 9	작업 3과 기계 2에 대한 시간 2	작업 5와 기계 1에 대한 시간 1
작업 1과 기계 4에 대한 시간 2	작업 3과 기계 3에 대한 시간 4	작업 5와 기계 2에 대한 시간 10
작업 1과 기계 5에 대한 시간 1	작업 3과 기계 4에 대한 시간 8	작업 5와 기계 3에 대한 시간 1
작업 2와 기계 1에 대한 시간 10	작업 3과 기계 5에 대한 시간 1	작업 5와 기계 4에 대한 시간 6
작업 2와 기계 2에 대한 시간 1	작업 4와 기계 1에 대한 시간 7	작업 5와 기계 5에 대한 시간 4
작업 2와 기계 3에 대한 시간 1	작업 4와 기계 2에 대한 시간 3	
작업 2와 기계 4에 대한 시간 2	작업 4와 기계 3에 대한 시간 3	

그리고 나면 곧이어 입력자료의 확인작업을 요구한다. 본 프로그램은 언제나 입력이 끝난 후에는 확인메시지가 디스플레이 된다.

《입력확인》

	기계 1	기계 2	기계 3	기계 4	기계 5
작업1	8	10	9	2	1
작업2	10	1	1	2	8
작업3	1	2	4	8	1
작업4	7	5	3	3	7
작업5	1	10	1	6	4

DATA를 수정하시겠습니까(Y/N)N

작업시간의 입력의 정확성을 “N”를 입력시킴으로써 확인하고 나면 이제는 각 작업에 대한 납기일을 입력하도록 한다.

《납기일의 입력》

작업 1 : 32
작업 2 : 35
작업 3 : 41
작업 4 : 46
작업 5 : 52

곧이어 납기일을 확인하도록 메세지가 나온다.

만일 작업시간 입력과정에서 2번째 기계의 2번째 작업의 작업시간을 1로 입력시켜야 하는데 11로 입력하였다
고 하면 작업시간 입력확인시에 기계 2와 작업2에서 다음과 같이 11을 발견할 수 있었을 것이다.

〈입력확인〉

	기계 1	기계 2	기계 3	기계 4	기계 5
작업1	8	10	9	2	1
작업2	10	11	1	2	8
작업3	1	2	4	8	1
작업4	7	5	3	3	7
작업5	1	10	1	6	4

그러면 다음의 데이터를 수정하시겠습니까 “Y”를 입력하고 나면 수정할 작업번호, 수정할 기계번호와 올바른 작업시간 순으로 차례로 질문이 나오게 되므로 그에 따라서 적절히 응답입력을 넣으면 되는 것이다.

DATA를 수정하시겠습니까(Y/N)N

수정할 작업번호는2

수정할 기계번호는2

올바른 작업시간은1

그리고 나면 다시 정정된 자료에 의한 입력확인 메세지가 다음과 같이 나오게 된다.

〈입력확인〉

	기계 1	기계 2	기계 3	기계 4	기계 5
작업1	8	10	9	2	1
작업2	10	1	1	2	8
작업3	1	2	4	8	1
작업4	7	5	3	3	7
작업5	1	10	1	6	4

DATA를 수정하시겠습니까(Y/N)N

모든 입력절차와 더불어 정확성의 확인 끝나면 다음 메세지와 함께 프로그램이 수행된다.

현재 SEQUENCE를 탐색중

곧이어 프로그램의 초기해인 작업순서가 출력된다. 본 예제에서의 작업순서는 3-5-2-1-4번 작업순서로
작업을 하도록 순위가 결정된 것이다. 그 다음 비용은 제시된 초기작업순서에 따른 재고비용과 지연비용이 출력된다. 본 예제에서는 761원이 된다.

〈초기 SEQUENCE〉

3-5-2-1-4

비용 : 761원

그 다음 본 프로그램은 다음과 같은 메세지를 CRT 상에 디스플레이 한다.

현재 더 나은 SEQUENCE 탐색중

이때 프로그램은 첫번째 출력된 작업순위를 차례로 하나씩 바꿔 가면서 비용을 계산하면서 더 나은 해를 찾는 것이다. 예제에서는 초기해가 3-5-2-1-4이므로 3번째 작업과 5번째 작업의 순위를 바꿔서 5-3-2-1-4 작업순위의 내용을 계산해 보고 그 다음에는 5와 2의 작업을 바꾸는 순서로 계산해서 비용을 계산해 가는 것이다. 참고로 각각의 순위에 대한 비용을 보면 다음과 같다.

순위	비용
5-3-2-1-4	796원
3-2-5-1-4	807원
3-5-1-2-4	771원
3-5-2-4-1	771원

예제에서는 초기해 보다 더 좋은 해를 발견할 수 없으므로 다음과 같은 메세지가 출력된다.

3-5-2-1-4

위보다 더 좋은 해를 찾을 수 없습니다.

최소비용은 761원입니다.

이것으로 본 문제는 끝났지만 감도분석이라든지 수치를 바꿔서 문제를 풀어보고 싶을 경우를 감안하여 다음과 같은 메세지가 나온다.

다른 문제를 하시겠습니까(Y/N)N

“N”을 입력하게 되면 프로그램은 끝나게 되지만 “Y”를 입력하게 되면 프로그램은 다시 최초로 돌아가서 데이터의 입력을 요구하게 된다.

4. FMS 프로젝트의 일정계획용 소프트웨어

FMS의 설비계획은 수십개 또는 그 이상의 설비를 순차적으로 도입하는 절차로 이루어져 있다. 그러므로 이를 효율적으로 도입 초기의 목적을 달성하기 위해서는 프로젝트의 일정계획을 합리적으로 작성할 필요가 있다. 이를 위해서는 기본적으로 PERT/CPM기법이 개발되어 있으므로 이를 프로그램하여 소프트웨어를 개발하였다.

본 프로그램에서 사용한 수치에는 그림 5와 같다.

프로그램을 수행시키면 다음과 같이 FMS 프로젝트를 실현시키기 위한 소프트웨어임을 알리는 메시지가 나오게 된다.

프로젝트형 FMS의 일정계획용 소프트웨어

한양대학교 산업공학과
공장자동화 시스템 연구실
전화 292-2111 (교)2306
교수 윤덕균

그림 5. PERT 네트워크 사례

1 0 0 1 5 3 0 2 5 5 1 0 3 6 6 7 9 4 13 13 7 0 6 20 20 8 4 3 5 5

그 다음 곧이어 PERT 네트워크에서의 총 요소공정수를 묻는 물음이 나타나게 된다. 사례에서는 6을 입력시키면 된다.

*총가지수는 얼마입니까 : 6

그 다음 각 요소공정(PERT에 있어서 각 가지)의 자료를 입력하도록 요구한다. 각 요소공정의 시작과 끝 단계, 즉 시발노드와 종착노드, 그리고 요소공정의 작업시간의 낙관치(T_0), 보통치(T_m), 비관치(T_p)를 입력시키게 된다. 사례문제에서는 6개의 요소공정이 있으므로 6번째 가지에 대한 입력을 다음과 같이 순서에 따라 하면 된다.

1번째 가지에 관한 입력

- * 시발노드 : 1
- * 종착노드 : 2
- * 시간입력
 - = 낙관치 : 2
 - = 보통치 : 5
 - = 비관치 : 8

2번째 가지에 관한 입력

- * 시발노드 : 2
- * 종착노드 : 3
- * 시간입력
 - = 낙관치 : 1
 - = 보통치 : 1
 - = 비관치 : 1

3번째 가지에 관한 입력

- * 시발노드 : 3
- * 종착노드 : 5
- * 시간입력
 - = 낙관치 : 0
 - = 보통치 : 6
 - = 비관치 : 18

4번째 가지에 관한 입력

- * 시발노드 : 1
- * 종착노드 : 4
- * 시간입력
 - = 낙관치 : 3
 - = 보통치 : 3
 - = 비관치 : 3

5번째 가지에 관한 입력

- * 시발노드 : 4
- * 종착노드 : 5
- * 시간입력
 - = 낙관치 : 2
 - = 보통치 : 8
 - = 비관치 : 14

6번째 가지에 관한 입력

- * 시발노드 : 5
- * 종착노드 : 6
- * 시간입력
 - = 낙관치 : 7
 - = 보통치 : 7
 - = 비관치 : 7

입력이 끝나고 나면 곧이어 PERT에 대한 모든 단계(노드)에 따른 最早時點시각과 最遲時點시각의 평균치와 각각의 분산 그리고 여유시각(slack)이 다음과 같이 출력된다.

PERT 나무

노드번호	최초시 평균	점시각 분산	최지시 평균	점시각 분산	여유시각
1	0	0	0	10	0
2	5	1	5	9	0
3	6	1	6	9	0
4	3	0	5	4	2
5	13	10	13	0	0
6	20	10	20	0	0

곧이어 프로젝트 기간의 평균치와 분산이 다음과 같이 출력된다.

*평균 PROJECT 기간 : 20

*PROJECT 기간의 분산 : 10

그 다음 프로젝트의 예정기일을 묻게 되는데 이는 곧이어 그 기간동안에 프로젝트가 완성될 확률을 계산하기 위한 것이다.

*PROJECT의 예정 완료기일은 열마입니까22

곧이어 프로젝트가 22기간내에 완성될 확률 .736이 출력된다.

PROJECT가 예정 기일에 완성될 확률 : .736

이것으로써 본 사례에 대한 컴퓨터 수행은 일차적으로는 끝나게 된 것이다. 감도분석 등 다른 문제를 위해서 본 프로그램에는 다음과 같은 메뉴가 다시 준비되어 있다.

*다음중 원하는 작업은 무엇입니까

1. 새로운 문제 수행
2. 예정완료시간 변경
3. 프로그램 종료

원하는 작업의 번호를 선택하십시오2

예정완료시간을 변경해서 다시 프로그램을 수행하고 싶을 경우에는 「2」를 입력시키면 된다. 그러면 곧이어 평균프로젝트의 기간, 기간의 분산, 그리고 예정완료기일에 대한 입력 요구가 다음과 같이 있게 된다.

*평균 PROJECT 기간 : 20

*PROJECT 기간의 분산 : 10

*PROJECT의 예정 완료기일은 얼마입니까?

위와 같이 예정완료기일에 17일을 입력하게 되면 다음과 같이 프로젝트가 17일안에 완성될 확률 0.171이 출력된다.

PROJECT가 예정기일에 완성될 확률

: .171

또 다른 문제를 위한 다음 불음이 다시 CRT상에 나타나게 되는데 문제를 더 이상 계속하고 싶지 않을 경우에는 다음과 같이 3을 입력하면 된다.

*다음중 원하는 작업은 무엇입니까?

1. 새로운 문제 수행
2. 예정완료시간 변경
3. 프로그램 종료

원하는 작업의 번호를 선택 하십시오.

5. 결 언

지금까지 FMS도입과 운영을 위한 소프트웨어를 개발하였으나 아직 모든 FMS타입에 가능한 소프트웨어의 개발에는 여부족인 실정이다. 그러나 FMS의 주류를 이루는 흐름작업형 FMS에 대한 해를 제시하였고 Jobshop형 FMS에서는 단일설비형 FMS의 dispatch 규칙을 이용해서 이를 보완 응용이 가능하리라 생각된다. 나머지는 차후 연구과제로 남긴다.

參 考 文 獻

- [1] Ashour, S. 1970. "A Branch-Bound Algorithm for Flow Shop Scheduling Problems," *AIEE Trans.*, 2, 172-176.
- [2] Ashour, S., T. E. Moore, and K. Y. Chiu. 1974. "An Implicit Enumeration Aigorithm for the Non-preemptive Shop Scheduling Problem," *AIEE Trans.*, 6, 62-72.
- [3] Baker, K. R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons.
- [4] Barnes, J. W., and J. J. Brennan. 1977. "An Improved Algorithm for Scheduling Jobs on Identical Machines," *AIEE Trans.*, 9, 25-31.
- [5] Campbell, H. G., R. A. Dudek and M. L. Smith. 1970. "A Heuristic Algorithm for the n-Job m-Machine Sequencing Problem," *Mgmt. Sci.*, 16, 630-637.
- [6] Charlton, J. M., and C. C. Death. 1970. "A generalized Machine Scheduling Algorithm," *Opnl. Res. Quant.*, 21, 127-134.
- [7] Coffman, E. G., Jr. 1976. *Computer and Job Shop Scheduling Theory*. New York: John Wiley & Sons.
- [8] Dannenbring, D. G. 1977. "An Evaluation of Flow Shop Sequencing Heuristics," *Mgmt. Sci.*, 23, 1174-1182.
- [9] Day, J. E., and M. P. Hottensten. 1970. "Review of Sequencing Research," *Naval Res. Logist. Quart.*, 17, 11-40.

- [10] Doll, C. L., and D. C. Whybark. 1973. "An Iterative Procedure for the Single Machine Multi-Product Lot Scheduling Problem," *Mgmt. Sci.* 20, 50-55.
- [11] Dorsey, R. C., T. J. Hodgson and H. D. Ratliff. 1974. "A Production Scheduling Problem with Batch Processing," *Opsns. Res.* 22, 1271-1279.
- [12] Eastman, W. L., S. Even and I. M. Isaacs. 1964. "Bound for the Optimal Scheduling of n Jobs on m Processor," *Mgmt. Sci.* 11, 268-279.
- [13] Eilon, S. 1978. "Production Scheduling." In *OR '78*, pp. 1-30, K. B. Haley(ed.) North-Holland, Amsterdam.
- [14] Elmaghraby, S. E., and S. Park. 1974, "On the Scheduling of Jobs on a Number of Identical Machines," *AIEE Trans.* 6, 1-13.
- [15] Emmons, H. 1969. "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness," *Opsns. Res.* 17, 701-715.
- [16] Garey, M. R., D. S. Johnson and R. Sethi. 1976. "Complexity of Flow Shop and Job Shop Scheduling," *Math. Opsns. Res.* 1, 117-129.
- [17] Gelders, L., and P. R. Kleindorfer. 1974. "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop;Part1. theory," *Opsns. Res.* 22, 46-60.
- [18] Gelders, L., and P. R. Kleindorfer. 1975. Coordinating Aggregate and Detailed Scheduling Decision in the One-Machine Job Shop;Part2. Computation and Structure," *Opsns. Res.* 23, 312-324.
- [19] Gere, W. S. 1966. Heuristics in the Job Shop Scheduling. *Mgmt. Sci.* 13, 167-190.
- [20] Gorzaler, T., And S. Sahin. 1978. "Flowshop and Jobshop Schedules : Complexity and Approximation," *Opsns. Res.* 26, 36-52.
- [21] Gupta, J. N. D. 1971. "An Improved Combinational Algorithm for the Flow-Shop Scheduling Problem," *Opsns. Res.* 19, 1753-1758.
- [22] Hitomi K. and I. Ham. 1976. "Operations Scheduling for Group Technology Applications," *Annals of CIRP*, 25, 419-421.
- [23] Hitomi, K. and I. Ham. 1977. "Machine Loading for Group Technology Applications," *Annals of CIRP*, 26, 279-281.
- [24] Hitomi, K. and I. Ham. 1977. "Group Scheduling Technique for Multiproduct Multistage Manufacturing System," *Journal of Engineering for Industry, Transactions of the ASME*, 99, 759-765.
- [25] Horn, W. A. 1973. "Minimizing Average Flow Time with Parallel Machines," *Opsns. Res.* 21, 864-847.
- [26] Ignall, E., and L.Schrage. 1965. "Application of the Branch-and-Bound Technique to Some Flow-Shop Scheduling Problems," *Opsns. Res.* 13, 400-412.
- [27] Jagannathan, R., and M. R. Rao. 1973. "A Class of Deterministic Production Planning Problems," *Mgmt. Sci.* 19, 1295-1300.
- [28] Johnson, S. M. 1954. "Optimal Two-Stage Production Schedules With Setup Times Included," *Naval Res. Logist. Quant.* 1, 61-68.
- [29] Lageweg, B. J., J. K. Lenster and A. H. G. Rinnooy Kan. 1977. "Job-Shop Scheduling by Implicit Enumeration," *Mgmt. Sci.* 24, 441-450.
- [30] Lageweg, B. J., J. K. Lenster and A. H. G. Ronnooy Kan. 1978. "A General Bounding Scheme for the Permutation Flow-Shop Problem," *Opsns. Res.* 26, 53-56.
- [31] Lawler, E. L. 1973. "Optimal Sequencing of a Single Machine Subject to Precedence Constraints," *Mgmt. Sci.* 19, 544-546.
- [32] Lenstra, J. K. 1976. *Sequencing of Enumerative Methods*. Mathematisch Centrum, Amsterdam.
- [33] McMahon, G. B., and P. G. Burton. 1967. Flow Shop Scheduling with the Branch-and-Bound Method," *Opsns. Res.* 15, 473-481.
- [34] McNaughton, R. 1959. "Scheduling with Deadlines and Loss Functions," *Mgmt. Sci.* 6, 1-12.

- [35] Moore, J. M. 1968. "An N Job One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs." *Mgmt. Sci.* 15, 102-109.
- [36] Nummikoven, T. S., and H. Emmons. 1977. "Scheduling on Parallel Machines to Minimize Two Criteria Related to Job Tardiness," *AIEE Trans.* 9, 288-296.
- [38] Panwalker, S. S., and W. Iskander. 1977. "A Survey of Scheduling Rules," *Opsns. Res.* 25, 45-61.
- [39] Park, K. and D. K. Yun. 1984. "A Stepwise Enumeration Algorithm for the Economic Scheduling Problem," *AIEE Trans.*, 16, 365-370.
- [40] Park, K. and D. K. Yun. 1984. "Optimal Scheduling of Periodic activities," *Opsns. Res.* 33, 690-695.
- [41] Rinnooy Kan, A. H. G., B. J. Lageweg and J. K. Lenstra. 1975. "Minimizing Total Costs in One-Machine Scheduling," *Opsns. Res.* 23, 908-927.
- [42] Root, J. G. 1965. "Scheduling with Deadlines and Loss Functions on k Parallel Machines," *Mgmt. Sci.* 11, 460-475.
- [43] Rothkopf, M. 1966. "Scheduling Independent Tasks on Parallel processors," *Mgmt. Sci.* 12, 437-477.
- [44] Saadov, M. S. 1978. "Scheduling and Sequencing," In *Handbook of Operations Research: Models and Applications*, II, pp. 268-300, J. Moder and S. E. Elmaghraby(eds.). Van Nostrand, Florence.
- [45] Sen, T. T., and L. M. Austin. 1980. An Efficient Algorithm for Minimizing Total Tardiness in the n-Job, One-Machine Sequencing Problem, Working Paper, College of Business Administration, Texas Tech University, Lubbock, Texas.
- [46] Shwimer, J. 1972a. "On the N-Job, One-Machine, Sequence-Dependent Problem with Tardiness Penalties : A Branch-and-Bound Approach," *Mgmt. Sci.* 18, B301-B313.
- [47] Smith, R. D., and R. A. Dudek. 1967. "A General Algorithm for Solution of the n-Job m-Machine Scheduling Problem of the Flow Shop," *Opsns. Res.* 15, 71-81. (and "Errata" *Opsns. Res.* 17, 756 [1969])
- [48] Srinivasan, V. 1971. "A Hybrid Algorithm for the One-Machine Sequencing Problem to Minimize Total Tardiness," *Naval Res. Logist. Quart.* 18, 317-327.
- [49] Stecke, K. E., and J. J. Solberg. 1977. "Scheduling of Operations in a Computerized Manufacturing System," *NSF Grant No. APR74 15256, Report*, 10, School of Industrial Engineering, Purdue University.
- [50] Yoshida T., Nakamura, N., and K. Hitomi. 1978. "Group Production Scheduling for Minimum Total Tardiness," *AIEE Trans.*, 10, 157-162.
- [51] Yun, D. K. and S. Park. 1983. "Product Selection and Scheduling on One Facility," *International Journal of Policy and Information*, 7, 79-82.
- [52] 大橋和正, 人見勝人, 「단일다능생산시스템의 공정계획과 스케줄링에 관한 연구」, 日本經營工學會誌, 33 (2), 1982, 106.
- [53] 岩田一明, 「흐름생산형 FMS의 일정계획」, 機構論, 810-4, 1981, 158.
- [54] 윤덕균, 생산관리용 소프트웨어집, 대영사, 1987.