FMC에서의 일정계획 시스템의 설계

Design of Scheduling System for Flexible Manufacturing Cells

신 대 혁*
이 상 완**

Abstract

The purpose of this paper is to describe an effective cell scheduling system for flexible manufacturing cells. Based on the FMC characteristics, cell scheduling can be defined as a dynamic modified flow shop working in a real-time system.

This paper attempt to find the optimal cell scheduling when minimizing the mean flow time for n-job/m-machine problems in static and dynamic environments. Real-time scheduling in an FMC environment requires rapid computation of the schedule.

I. 서론

FMC(Flexible Manufacturing Cell)는 FMS(Flexible Manufacturing System ; 유연생산시스템)를 구축하고자 하는 기업에서 시스템의 대형화에 따라 발생할 수 있는 투자비용에 대한 위험부담을 줄이고자 하는 이유와, 자본제약으로 인해 셀 단위의 자동화를 구현하기 위해 널리 인식되고 있다[1]. 또 FMC는 GT기법을 통한 기계 셀의 형성과 셀 내에서 처리되는 부품군 형성을 토대로 셀 내에 들어오는 작업대상물의 작업들에 대한 정보를 사전에 컴퓨터에 기억시켜두고, 생산일정계획에 따라서 작업대상물을 운반가공하는 자동화된 생산 시스템이라 할 수 있다.

FMS와 FMC간의 일반적인 구별은 기계군 형성시에 기계의 수에 따라 여러가지로 구별하고 있으나, 전형적인 FMC는 3-4대의 기계로 구성되는데, 검사장비와 대개 로보트에 의해 운용되는 내부 물류장비가 포함된다[2].

일정계획에 관한 대부분의 연구는 총가공시간(makespan)을 최소화 하는 것을 목적으로 하고 있으며, Johnson's rule(존슨기법)의 개발과 이 기법에 기초한 응용들이 많이 이루어지고 있다[3]. 그러나 흐름시간(flow time)에 관한 연구는 많이 행해지지 않고 있으며, 이 분야의 대표적인 연구로는 "2대의 기계"와 "복수-작업" 문제를 해결하기 위해 제안된 분지한계법(branch and bound procedure)이 있다[4,5]. 또한, 항상 최적해를 제공해 주지는 않지만, 어느 정도 적용 가능한 몇몇 발견적 기법(heuristic method) 들이 개발되어 졌으나[3,6,7,8,9], 휴리스틱 기법 대부분이 그렇듯 이들 알고리즘들은 특정 사례를 위해서 개발된 관계로 계산시간과 특정한 적용조건으로 인해 실시간에 적용하기가 어렵다는 문제가 있다.

본 연구는 Campbell 알고리즘[3] 및 Gupta 알고리즘[6]과 본 연구에서 제안한 방법간의 효율성을 비교하였다. 그리고 FMC의 특성에 기초를 둔 셀 일정계획은 실시간에서의 동적 수정흐름작업 연구로 정의되고 있으나, 정적 일정계획에서 보다 많은 연구가 행해지고 있다.

따라서 본 연구는 정적, 동적 환경에서 n-작업/m-기계의 평균흐름 시간을 최소화 하려는 셀 일정계획에 대해 다루었으며, 보다 신속한 셀 일정계획을 수립하였다.

* 마산전문대학 산업안전관리과
** 동아대학교 산업공학과

II. 셀 일정계획의 모델

1. 셀 일정계획의 분류

본 연구에서는 셀 일정계획 문제를 다음과 같이 정적 셀 일정계획과 동적 셀 일정계획으로 분류하여 문제를 다루고자 한다.

① 정적 셀 일정계획 (Static cell scheduling problem)
작업그룹이 셀에 도착했을때 작업순서를 결정하며, 대개 표준치구와 파레트를 이용한다.

② 동적 셀 일정계획 (Dynamic cell scheduling problem)
셀에서 초기기계가 이용가능한 상태가 될 때 작업순서를 결정하며, 동적 셀 일정계획의 목적은 평균흐름 시간을 최소화하거나 기타 다른 일정계획 수행도 기준 (기계 이용도, 평균처리시간, 총가공시간)을 최소화 하는 것이다.

2 전제 조건

가. 기본가정

① 작업가공시간은 확정적이고 사전에 알려져 있다. 공정처리 장비는 자동이다.
② 한 기계에서 한번 시작된 작업은 완성하기 前까지는 중단될 수 없다.
③ 작업은 엄격하게 요구된 가공순서에 따라 구성된다.
④ 입력 파레트는 같은 부품군을 가지며, 파레트가 부품으로 채워지고 기계가 이용가능한 상태가 되는 즉시 또 다른 파레트가 셀에 도착한다.
⑤ 같은 작업순서는 같은 부품군으로 가정한다.
⑥ 기계는 생산기간 동안 언제라도 積荷가 가능하다.(준비 작업은 특정 운반에 제한을 받지 않는다.)

나. 부수적 가정 (동적 셀 특성에 따른 제약조건)

① 부품들은 파레트의 여유 저장공간이 발생할 때마다 셀에 계속적으로 동적으로 들어온다.
② 작업순서는 최초의 기계가 셀에서 이용가능하여 졌을 때 결정된다.
③ 가공 및 물류장비는 네트워크 통신을 통한 셀 제어기에 의해 제어되고, 모든 관련 가공 데이타는 언제든지 검색 가능하도록 데이타 베이스에 저장된다.

III. 모델의 설정과 발견적 기법

1. 정적모델 (Static model)

본 연구에서 개발한 발견적 기법은 모든 작업에 대한 조합을 고려하기 보다는 작업쌍에 대해서만 평가하였다. 이러한 알고리즘은 기계의 일정계획에 작업을 부여할 경우, 작업쌍의 평가를 이용하는 것이 최소 총 작업흐름 시간을 구하기가 빠르며, 보다 좋은 최적해를 가져다 줄 것이다.

우선작업 i가 작업 j에 선행되는 2개의 작업, 2대의 기계를 예를 들어 설명하고자 한다. Figure 1은 2대의 기계문제에 관한 간트차트(Gantt chart)이며. 두 가지 흐름시간 순서는 다음과 같다.

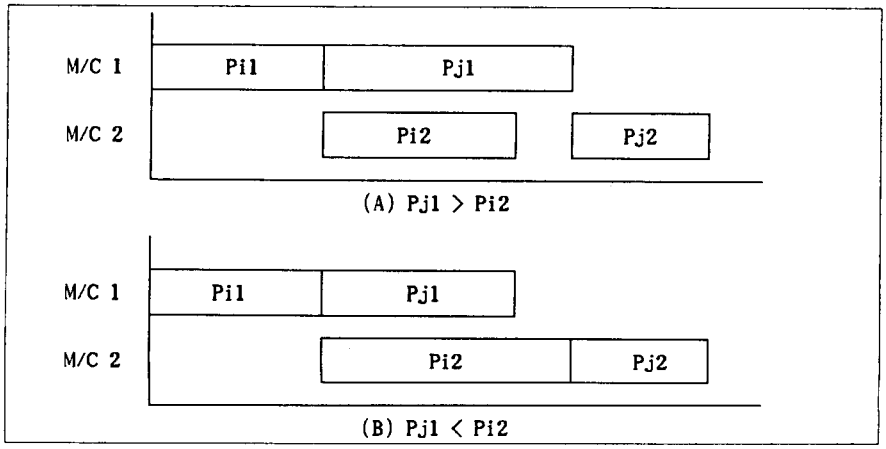

Figure 1. 2대의 기계 문제에 관한 간트 차트

Figure 1(A)는 작업 i의 흐름시간이 (P_{i1} + P_{i2})이고, P_{iK}는 기계 K에서 작업 i를 처리하는데 걸리는 시간을 나타낸다. 마찬가지로 작업 j에 대한 흐름시간은 (P_{i1} + P_{j1} + P_{j2})이고, 두 작업의 총 흐름시간은 식 (1)과 같다.

$$F_{2(ij)} = 2P_{i1} + P_{i2} + P_{j1} + P_{j2} \quad \cdots\cdots (1)$$

여기서, $F_{m(ij)}$는 m대의 기계에서의 작업 i와 j의 총흐름시간을 나타내며, 작업 i는 작업 j에 선행된다. Figure 1. (B)에서의 총흐름시간은 식 (2)와 같이 정의된다.

$$F_{2(ij)} = 2P_{i1} + 2P_{i2} + P_{j2} \quad \cdots\cdots (2)$$

식 (1)과 식 (2)는 실행가능해이기 때문에, 최소 총흐름시간은 식 (3)과 같다.

$$F_{2(ij)} = 2P_{i1} + P_{i2} + P_{j2} + \max(P_{j1}, P_{i2}) \quad \cdots\cdots (3)$$

만약, 작업 j가 작업 i에 선행된다면, 식 (3)은 식 (4)와 같이 된다.

$$F_{2(ji)} = 2P_{j1} + P_{j2} + P_{i2} + \max(P_{i1}, P_{j2}) \quad \cdots\cdots (4)$$

따라서, 총 흐름시간이나 평균흐름시간을 최소화하는 순서를 결정하면 식(5)와 식 (6)이 된다.

$$F^{*}_{2(ij)} = 2P_{i1} + \max(P_{j1}, P_{i2}) \quad \cdots\cdots (5)$$
$$F^{*}_{2(ji)} = 2P_{j1} + \max(P_{i1}, P_{j2}) \quad \cdots\cdots (6)$$

2대의 작업, 3대의 기계문제에 관한 순서는 Figure 2와 같다.

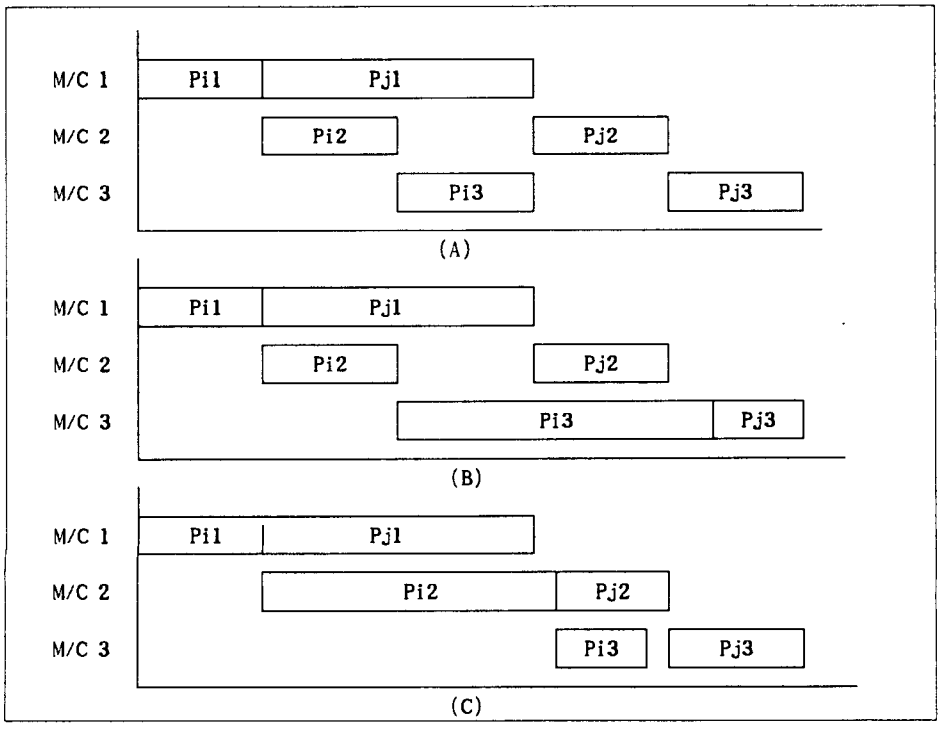

Figure 2. 3대의 기계 문제에 관한 간트차트

P_{i1}은 순서에서 가장 먼저고 P_{j3}은 가장 마지막이기 때문에, 최소 흐름시간은 다음 작업들의 최대 작업순서에 영향을 받는다.

$$(P_{j1} + P_{j2}) \text{ 또는 } (P_{i2} + P_{i3}) \text{ 또는 } (P_{i2} + P_{j2})$$
$$\longrightarrow \max [P_{j2} + \max(P_{j1},P_{i2}),\ P_{i2} + P_{i3}]$$

따라서, 총 흐름시간은 식 (7)과 같다.

$$F_{3(ij)} = 2P_{i1} + (P_{i2} + P_{i3}) + P_{i3} + \max[P_{j2} + \max(P_{j1},\ P_{i2}),\ P_{i2} + P_{i3}] \quad \cdots\cdots\cdots (7)$$

식 (7)과 같은 요령으로 $F_{3(ji)}$를 구한뒤, 공통항을 제거하면 다음과 같은 일반형을 구할 수 있다.

$$F^{*}_{3(ij)} = 2P_{i1} + P_{i2} + \max[P_{j2} + \max(P_{j1},\ P_{i2}),\ P_{i2} + P_{i3}]$$

같은 방식으로 4대나 5대의 기계에 관한 식을 유도할 수 있으며, m대의 기계에 관한 일반식을 구하는 과정이 식 (8)에서 식 (11)로 나타낼 수 있다.

$$F_{m(ij)} = 2P_{i1} + \sum_{k=2}^{m} P_{ik} + R_m \quad \cdots\cdots\cdots (8)$$

$$F^{*}_{m(ij)} = 2P_{i1} + \sum_{k=2}^{m-1} P_{ik} + R^{*}_{m} \quad \cdots\cdots\cdots (9)$$

여기서 $R_1 = 0$

$R_2 = P_{j2} + \max(P_{j1},\ P_{i2})$

$R_3 = P_{j3} + \max(R_2,\ P_{i2} + P_{i3})$

$R_4 = P_{j4} + \max(R_3,\ P_{i2} + P_{i3} + P_{i4})$

. . . .

. . . .

$$R_m = P_{jm} + \max(R_{m-1},\ \sum_{k=2}^{m} P_{ik}) \quad \cdots\cdots\cdots (10)$$

$$R^{*}_{m} = R_m - P_{jm} \qquad m \geq 2 \quad \cdots\cdots\cdots (11)$$

2. 정적 셀 일정계획의 알고리즘

작업순서는 다음과 같은 단계로 실시된다.

Step 1 ; 셀 데이타로 부터 처리시간과 작업경로 정보를 검색한다.

Step 2 ; 식 (9)를 이용하여 작업들의 단순화된 잠정흐름시간을 계산한다. 예를들어, 3개의 작업, m대의 기계문제의 경우, 6개의 잠정흐름시간 $F^{*}_{m(1,2)}$, $F^{*}_{m(21)}$ $F^{*}_{(13)}$, $F^{*}_{m(31)}$, $F^{*}_{m(23)}$, $F^{*}_{m(32)}$를 구한다.

Step 3 ; 잠정흐름시간을 계산하고 가장 작은 값을 찾아낸다. 모든 흐름시간 쌍들에 대해 구한 뒤 그 순서쌍의 시작 작업에 *표시를 한다. *의 갯수를 세어서 * 의 갯수가 제일 많은 것을 가장 앞선 작업으로 선택한다.

3. 동적 모델 (Dynamic model)

정적모델은 작업이 셀에 할당되었을 때 기계는 선행작업을 끝마치고 쉬고 있는 상태라고 가정한다. 반면에 동적모델은 새로운 작업이 셀에 도착했을 때 기계는 아직 선행작업을 처리하고 있으며, 일정계획시 현재의 이러한 셀 상태를 고려하여야만 한다.

따라서, 동적 셀 모델은 셀 내에 작업하고 있는 이들 작업들의 잔여처리 작업들을 포함한다. 근본적으로는 정적 셀 일정계획모델과 동적 셀 일정계획은 똑 같다. 예를들어, 2개의 작업, 2대의 기계상황을 고

려해 보자. 새로운 작업이 일정계획하에 고려되었을 때, 기계 h에서 작업 g를 처리하기 위해 기다려야 하는 시간을 Q_{gh}라 정의하자. 이때 4개의 순서관계는 Figure 3과 같다.

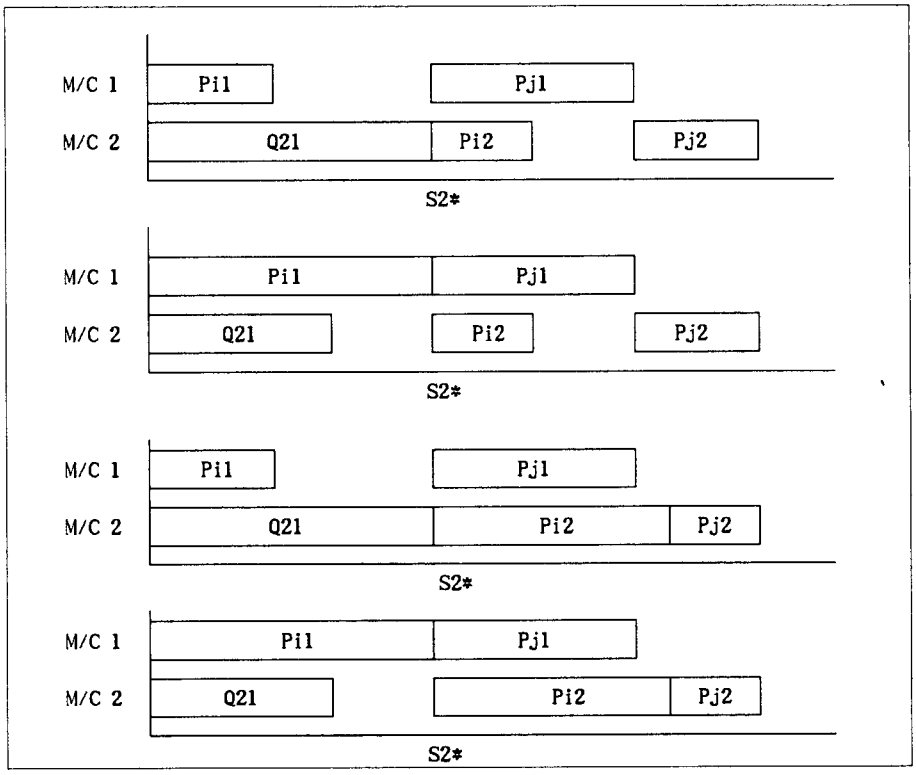

Figure 3. 2개의 부품, 2대의 기계 경우에 있어서 처리시간과 잔여 처리시간

만약 작업 i가 작업 j의 선행작업이라면(Figure 3에서 i=1, j=2), 그 때 최소 총 흐름시간은 식 (12)와 같다.

$$F_{2(ij)} = 2 * \max(P_{i1}, Q_{21}) + P_{i2} + P_{j2} + \max(P_{j1}, P_{i2}) \quad \cdots\cdots \quad (12)$$

작업 j가 작업 i의 선행작업이라면, 최소 총 흐름시간은 식 (13)과 같다.

$$F_{2(ji)} = 2 * \max(P_{j1}, Q_{21}) + P_{j2} + P_{i2} + \max(P_{i1}, P_{j2}) \quad \cdots\cdots \quad (13)$$

식 (12)와 식 (13)을 정적모델과 같은 방법으로 잠정 흐름시간을 구하면 식 (14)와 식(15)와 같다.

$$F^*_{2(ij)} = 2 * \max(P_{i1}, Q_{21}) + \max(P_{j1}, P_{i2}) \quad \cdots\cdots \quad (14)$$

$$F^*_{2(ji)} = 2 * \max(P_{j1}, Q_{21}) + \max(P_{i1}, P_{j2}) \quad \cdots\cdots \quad (15)$$

만약, 여기서 식 (14)와 식 (15)에 ($Q_{21} = 0$)이 할당되면, 식 (5)와 식 (6)의 정적 모델과 같아진다.

일반적인 잠정흐름시간 공식은 정적모델의 경우와 같은 방법으로 구하면 되지만, 본 연구에서는 4대의 기계까지만 실행가능하다. 따라서, 여기서는 3대의 기계, 4대의 기계에 관한 일반형만 나타내고자 한다.

$$F^*_{m(ij)} = 2S^*_m + T^*_m \quad \cdots\cdots \quad (16)$$

여기서

$$T^{*}_{m} = [\max(A_m + P_{j(m-2)} - S^{*}_{m}) + P_{j(m-1)}, P_{im}] \quad \cdots\cdots\cdots\cdots \quad (17)$$

$A_2 = 0$

$A_3 = \max(P_{i1}, Q_{21})$

$A_4 = \max[\max(P_{i1}, Q_{21}) + P_{i2}, \max(P_{21}, Q_{31}), P_{31}]$

$B_3 = \max(Q_{21}, Q_{31})$

$B_4 = \max[\max(Q_{21}, Q_{31}), \max(Q_{31}, Q_{41}) + Q_{32}]$

IV. 사례연구

1. 정적 셀 일정계획의 예

4개의 작업, 4대의 기계문제를 고려하자. 각각의 처리시간은 Table 1과 같다.

Table 1. 4개의 작업/4대의 기계문제의 처리시간

job	machine			
	1	2	3	4
1	22	11	19	21
2	9	14	16	2
3	20	19	4	2
4	10	18	6	7

우선 각 순서쌍들의 잠정흐름시간을 구하면 Table 2와 같다.

Table 2. 순서쌍과 대응되는 잠정흐름시간

pairwise	P*
1 - 2 ; 2 - 1	125 ; 100
1 - 3 ; 3 - 1	125 ; 115
1 - 4 ; 4 - 1	125 ; 96
2 - 3 ; 3 - 2	91 ; 112
2 - 4 ; 4 - 2	86 ; 92
3 - 4 ; 4 - 3	106 ; 87

예를 들어, 순서 1 - 2는 식 (9)를 이용하여 구하면 다음과 같다.

$$F^{*}_{4(12)} = 2(22) + (11 + 19) + 51 = 125$$

여기서, 첫번째 항은 P_{11}이고 두번째 항은 $P_{12} + P_{13}$이고, 세번째 항은 다음과 같다.

$\max\{R_3, 51\}$

$= \max\{16 + \max[R_2, 30], 51\}$

$= \max\{16 + \max[14 + \max(9, 11), 30], 51\}$

$= \max\{16 + \max[(14 + 11), 30], 51\}$

$= \max\{(16 + 30), 51\}$

$= 51$

순서 2 - 1에 대해서는,

$$F^{*}_{4(21)} = 2(9) + (14 + 16) + 52 = 115$$

이고, 세번째 항 52는 다음과 같이 구한다.

$\max\{19 + \max[11 + \max(22, 14), 30], 32\}$

$= \max\{19 + \max[(11 + 22), 30], 32\}$

$= \max\{(19 + 33), 32\}$

$= 52$

$F^*_{4(21)}$이 $F^*_{4(12)}$ 보다 작기 때문에 $F^*_{4(21)}$의 시작 작업 2에 *표시를 한다.
모든 순서쌍들에 대한 계산은 Table 3과 같다.

Table 3. 잠정흐름시간으로 부터 순서결정

pairwise	F*
1 - 2 ; 2 - 1	2* - 1
1 - 3 ; 3 - 1	3* - 1
1 - 4 ; 4 - 1	4* - 1
2 - 3 ; 3 - 2	2* - 3
2 - 4 ; 4 - 2	2* - 4

(*표를 합하면 작업 2는 3개, 작업 4가 2개, 작업 3이 1, 작업 1은 0다. *의 순서에 따른 순서는 2-4-3-1이고, 평균흐름시간(최적)은 69.75가 된다.)

이 문제에서 만약 기계에 대한 작업들의 모든 조건을 계산한다면 24개의 가능성(4!)이 된다. 따라서, 이러한 발견적 순서쌍 방법을 이용하면 단지 6개의 조합으로 구할 수 있다.

2. 계산결과

가. 정적 셀 일정 계획

효율성과 최적성을 평가하기 위해 두가지 방법을 이용한다. 첫째는 모든 기계의 순서와 일치하는 순수 흐름작업에 대한 문제에 적합시켰으며, 수행도의 평가는 본 연구에서 제시한 알고리즘의 최적해와 다른 현존 알고리즘들과 비교하였다. 두번째는 셀 내에서 서로 다른 부분흐름시간을 갖는 변경 흐름작업에 적합시켜 보았다.

그리고 이 두가지 평가방법은 잠정흐름시간들을 구하는 연산방법을 고려하여, 최적잔존작업(MWKR ; most work remaining)과 최소잔존작업(LWKR ; Least work remaining)과 같은 일반적 일정계획 연산과 비교하였다.

수행도를 평가하기 위해 셀 구성도, 작업수, 기계대수와 같은 셀 구성운영 조건하에서 시뮬레이션 모델을 사용하였으며, 이러한 것들을 Code화 하여 FORTRAN 77로 실행 하였다.

그리고 실행조건은 작업수는 세 작업에서 여섯 작업까지이며, 기계수는 4대에서 8대까지 그 대상으로 하였으며, 반복횟수는 50회 실시하였다.

셀 구성도는 Figure 4와 같이 순수 흐름작업과 두가지 변경 흐름작업으로 제한을 두었다.

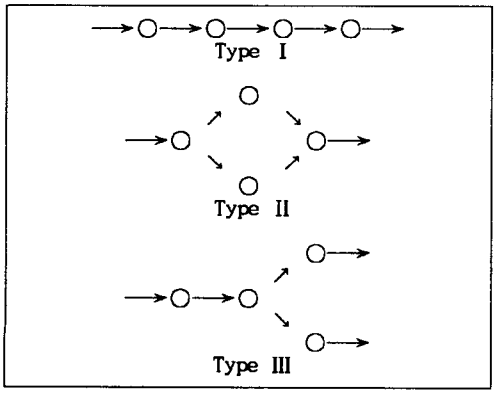

Figure 4. 셀 흐름 구성 형태

이와 같은 조건하에서 실행한 결과는, 첫째 제안된 알고리즘은 Gupta 알고리즘과 Campbell 알고리즘보다 수행도가 높다 (Table 4), 둘째 제안된 알고리즘은 계산 및 비교횟수가 상당히 적음을 알 수 있다

(Table 5), 셋째 Figure 4의 Type I이 Type II, III보다 최적흐름시간 비율면에서 우수한 해를 가져다 준다.

Table 4. 알고리즘들 간의 수행도 비교 [6]

(n)	(m)	Proposed algorithm (R)	Gupta* algorithm (R)	Campbell* algorithm (R)
3	4	0.993		
3	6	0.996		
3	8	0.996		
4	4	0.990	0.980	0.940
4	6	0.990	0.960	0.940
4	8	0.990	0.970	0.960
5	4	0.980	0.960	0.940
5	6	0.980	0.950	0.930
5	8	0.980	0.960	0.930
6	4	0.990	0.950	0.910
6	6	0.970	0.950	0.920
6	8	0.980	0.950	0.930

* n = 작업수 m = 기계대수 R ; 최적비율(optimal/actual)

Table 5. n-부품, m-기계문제의 실행횟수

Number of parts (n)	Proposed method (A)	Complete enumeration (B)	Ratio of A/B
2	1	2	2
3	3	6	2
4	6	24	4
5	10	120	12
6	15	720	48
7	21	5040	240
8	28	40320	1440
9	36	362880	10080
10	45	3628880	80640

Table 6. 셀 구성 형태에 따른 최적흐름시간 비율

n	m	Configuation type	Rate of optimality
3	4	I	0.993
3	4	II	0.981
3	4	III	0.990
4	4	I	0.987
4	4	II	0.965
4	4	III	0.971
5	4	I	0.982
5	4	II	0.956
5	4	III	0.928
6	4	I	0.986
6	4	II	0.938
6	4	III	0.936

* n = 작업수 m = 기계대수 반복횟수 = 50

나. 동적 셀 일정계획

동적 셀 일정계획 수행도의 결과는 LWKR기법과 비교하였을 때 더 우수한 알고리즘으로 판명되었으며, LWKR기법은 MWKR기법이나 SPT기법 보다 우수한 알고리즘으로 알려져 있다.

그리고 Table 7은 동적 알고리즘을 정적 알고리즘과 LWKR기법 간에 비교한 자료이다.

Table 7. 정적 및 동적 연산과 LWKR연산간의 비교

n	m	Static algorithm F	Static algorithm U	Dynamic algorithm F	Dynamic algorithm U	LWKR algorithm F	LWKR algorithm U
3	2	149.3	0.87	143.7	0.88	151.6	0.86
		161.2	0.84	153.3	0.86	162.1	0.84
4	2	167.1	0.86	157.6	0.91	174.4	0.85
		176.9	0.87	168.7	0.90	183.7	0.85
5	2	179.8	0.86	171.7	0.89	187.8	0.85
		188.3	0.87	175.5	0.90	192.3	0.86
3	3	205.0	0.78	195.1	0.78	208.2	0.78
		220.1	0.76	196.7	0.77	218.6	0.75
4	3	230.0	0.78	218.5	0.79	235.0	0.77
		221.3	0.78	203.7	0.79	225.4	0.77
5	3	259.7	0.77	244.5	0.79	264.1	0.77
		219.4	0.78	207.4	0.79	224.0	0.77
3	4	274.1	0.77	261.7	0.78	263.9	0.78
		268.1	0.76	251.9	0.77	256.0	0.76
4	4	302.2	0.78	284.9	0.78	293.8	0.78
		257.7	0.76	245.1	0.76	255.6	0.75
5	4	322.7	0.77	303.9	0.77	314.0	0.77
		303.5	0.77	284.9	0.77	290.8	0.76

* F = 평균흐름시간 U = 기계이용시간 n = 작업수 m = 기계대수

V. 결 론

본 연구에서 제안한 알고리즘은 정적 셀 일정계획에 있어서 Gupta나 Campbell 알고리즘과 비교하여 우수하고, LWKR기법과 비교하여서도 우수한 것으로 나타났다.

그리고 계산실행 횟수면이나 비교횟수에서도 제안된 알고리즘이 분지한계법, 순열 알고리즘, 동적계획법 등에 비해 적은 것으로 나타났다. 따라서, 실시간 FMC일정계획에 적용하기가 용이하며 보다 효과적이다.

앞으로의 연구과제는 동적모델에서 4대의 기계까지만 실행가능한데 비하여, 그 이상의 기계에 대해서도 적용이 가능할 수 있도록 연구할 필요가 있다.

參 考 文 獻

1. Snader,K.R., Flexible manufacturing systems : an industry overview. Production and Inventory Management, Vol.27, 1986.
2. Groover,M.P., Automation, Production systems, and Computer integrated manufacturing, Prentice-Hall International Editions, 1987.
3. Campbell,H.G., Dudeck,R.A., and Smith,M.L., "A heuristic algorithm for the n-job, m-machine seguencing problem," Management Science, Vol.16, 1970.
4. Bansal,S.P., "Minimizing the sum of completion times of n-jobs over m-machines in a flow shop - a branch and bound approach," AIIE Transactions, Vol.9, 1977.
5. Ignal,E., and Schrage,L., "Applications of the branch and bound technique to some flow-shop scheduling problems," Operations Research, Vol.26, 1965.
6. Gupta,J.N., "Heuristic algorithms for the multistage flow-shop scheduling problem," AIIE Transactions, Vol.4, 1972.
7. Krone,M., "Heuristic Programming solution of a flow-shop scheduling problem," Operations Research, Vol.22, 1974.
8. Miyazaki,S., Nishiyama,N., and Hashimoto,F., "An adjacent pairwise approach to the mean flow-time scheduling problem," J. of the Operations Research Society of Japan, Vol.21, 1978.
9. Sarin,S.C., and Eybl,D., "The two-machine mean-flow time flow shop problem and some special cases," TIMS - ORSA Meeting, Los Angles, November, 1978.