

動的 환경에서 flowshop 작업순서 決定을 위한 發見的 技法들의 有效性에 관한 연구

(An Evaluation of Flowshop Scheduling Heuristics in a Dynamic Environment)

박 양 병*

Abstract

This paper provides an evaluation of static flowshop scheduling heuristics for minimizing makespan as an objective function in the dynamic flowshop model, in which new jobs with stochastic processing times arrive at the shop randomly over time and are added into the waiting jobs for processing. A total of sixteen scheduling heuristics, including several revisions and combinations of previously reported methods, are summarized. These scheduling rules are evaluated via computer using a SLAM discrete event simulation model. The results for the simulation are analyzed using both statistical and nonstatistical methods. The results from the study suggest which of the popular scheduling rules hold promise for application to practical dynamic flowshop problems.

1. 서 론

Flowshop은 몇가지 엄격한 제약이 추가된 jobshop 형태이다. 즉, 모든 job들은 여러 대의 기계로 구성된 多階段工程을 거치며, 이들 job의 작업순서는 모든 기계들에 대해 동일하다. 일반적인 flowshop 문제에 주어진 조건들은 다음과 같이 요약된다[1].

(1) N개의 job들에 대한 가공은 M대의 기계를 일정한 순서로 거치며 이루어진다.

(2) Job의 가공을 위한 준비시간은 job의 작업순서에 대해 독립적이며, job가공시간에 포함된다.

(3) 기계에 대한 job의 가공시간, 준비시간, 그리고 가공방법에 대한 정보는 주어진다.

(4) M대의 기계는 job의 가공을 위해 항상 사용가능하다.

(5) 가공중인 job은 가공이 끝날때까지 기계에서 제거하지 못한다.

Flowshop 문제는 job의 가공을 받기 위한

*慶熙大學校 工科大学 産業工学科

대기상태와 가공시간의 정의에 따라 靜的 flowshop과 動的 flowshop 문제로 구별된다. 靜的 flowshop 문제에서는 모든 job들이 작업장에 이미 도착하여 시간 0에 가공이 가능하며, job가공시간은 확정적으로 정의된다. 반면에, 動的 flowshop 문제에서는, job이 일정한 시간동안 확률적 분포를 따르며 계속적으로 작업장에 도착하여 기계의 이용상태에 따라 가공을 받기 위해 대기한다. Job가공시간은 확률적 분포를 따른다.

Job의 작업순서는, 모든 job의 가공에 대한 정보를 종합적으로 포함하는 有効性 측정치 (Performance measure)의 선택에 근거하여 결정된다. 측정치는 일반적으로 job의 가공 완료시간의 함수로써 정의되는데, 총처리시간 (Makespan)이 널리 사용되고 있다. 이외에도 평균처리시간 (Mean flow time), 평균 job 늦음 (Mean job tardiness), 총기계 이용도 (Total machine utilization), 평균 반제품 재고량 (Average in-process inventory), 그리고 평균작업량 (Average work content)등을 작업순서에 대한 유효성 측정치로 사용하고 있다.

小규모 靜的 flowshop 문제에서 총처리시간을 최소화하기 위한 최적작업순서를 결정하기 위하여 Little et al. [2], Lomnicki [3], Bestwick 와 Hastings [4]에 의해 分岐限界法 (Branch-and-Bound 기법)의 적용이 시도되었다. 그러나 문제의 규모가 증가함에 따라 소요되는 계산시간이 지수적 (exponentially)으로 증가한다는 사실이 밝혀짐에 따라 이 기법의 적용은 小규모의 문제에 한정되어왔다.

Littger [5]는 $N \times M$ 靜的 flowshop 작업순서 결정문제를 위한 새로운 최적해법을 개발했다. 이 기법에서는, job가공시간들을 상호 교환하는 과정을 통해 최적 작업순서를 결정하고 있다. Baker와 Schrage [6]는 job의 가공에 대해 先後 제약을 고려한 최적작업순서를

결정하기 위하여 動的 計劃法 (Dynamic programming)의 적용을 시도하였다. 이러한 기법들 역시 계산상의 난이성과 비경제성에 기인하여 小규모 문제에 그 적용이 한정되어왔다.

따라서 최적해는 얻지 못하지만, 적은 계산량으로 빠른 시간내에 小규모 문제는 물론 대규모 문제에 대한 근사최적해를 얻는 많은 發見的기법 (Heuristic)들이 개발 소개되었다. 이 기법들의 공통점은, 실제 작업장에서 예견되는 動的 및 확률적 상황이 전혀 고려되어 있지 않은, 靜的 flowshop 문제를 가정하여 연구 개발 되었다는 것이다.

본 연구에서는 靜的 flowshop 문제의 총처리 시간을 최소화 하기 위한 작업순서 결정에 대한 여러 발견적 기법들을 고찰하고 또 새로운 발견적 기법들을 제안하여, 動的 flowshop 모델에서 이 기법들의 수행도와 효율성에 근거한 유효성을, SLAM (Simulation Language for Alternative Modeling)을 사용한 컴퓨터 전산모의 실험 (Computer Simulation)에 의해 비교 평가하고자 한다.

2. 발견적 기법의 고찰 및 개발

지난 20여년 동안, 총 처리시간을 최소화하기 위한 靜的 flowshop 작업순서 결정문제에 관한 많은 발견적 기법들이 소개되었다. 이 기법들의 기본적 개념은 아래 3종류로 구별될 수 있다.

(1) Johnson의 2대의 기계에 대한 기법 [7] 이용

(2) Job 가공시간의 기울기 指數 (Slope index) 이용

(3) 소요되는 기계들의 총 유휴시간 (Total machine idle time)의 최소화.

표1은 본 연구에서 고려되는 총 16가지 발견적 기법의 기본적 개념과 그 적용과정에서 사출되는 중 작업순서수를 나타내고 있다. 이들 16

가지의 발견적 기법중 12가지는 이미 문헌에 소개되어 있으며, 나머지 4가지는 본 연구과정에서 새로이 제안된 것이다.

2.1 발견적 기법의 문헌고찰

문헌조사에 의한 발견적 기법을 간단히 요약설명한다.

(1) Petrov 발견적 기법 (PETRO1)

1966년 Petrov [8]에 의해 개발되었으며, 원래의 M 대 기계문제를 가상의 2대 기계문제로 변환시켜서 Johnson의 기법을 적용하였다.

(2) Gupta 발견적 기법 (GUPTA)

1971년 Gupta [9]는 Johnson의 2대와 3대의 기계에 대한 분류함수(Sorting function)

를 근거로 M 대 기계문제를 위한 새로운 함수를 개발 사용하였다.

(3) Dannenbring 발견적 기법(RAP)

1977년 Dannenbring [10]은 Rapid Access Procedure (RAP)라 불리는 새로운 기법에서, 기계의 가공순서에 따른 加重值를 이용하여 원래의 M 대 기계문제를 가상의 2대 기계문제로 변환시켜 Johnson의 기법을 적용하였다.

(4) 근접탐색을 이용한 Dannenbring의 발견적 기법 (RACS)

1977년 Dannenbring [10]은 그의 RAP에 하나의 개선과정을 도입하였다. 즉, RAP의 적용에 의해 얻어진 해를 초기해로 하여 작업

Table 1. General Description of Heuristics Mentioned in Research

Heuristic No. / Identification*	Basic idea of sequencing		procedure	Total machine idle time	The number of sequences generated
	Johnson's algorithm	algorithm			
	Single	Multiple	Slope index		
1. PETRO1	×				2
2. PETRO2	×		×		2
3. GUPTA			×		1
4. RAP	×		×		1
5. RACS	×		×		m - 1
6. CAMP1		×			m - 1
7. CAMP2		×	×		m - 1
8. CAMP3		×			m - 1
9. PALMER			×		1
10. BONNE1			×		1
11. BONNE2	×		×		1
12. SPAC1				×	n
13. SPAC2				×	n
14. GELDER				×	1
15. NEH					$(n(n+1)/2) - 1$
16. DELTA				×	1

*abbreviated names for heuristics in research

순서에서 서로 인접한 job들의 순서를 상호 교환하는 반복적인 과정을 통하여 총 처리시간의 개선을 기도하였다.

(5) Campbell, Dudek, 그리고 Smith의 발견적 기법 (CAMP 1)

1970년 Campbell 외 3인 [11]은, 일반적인 M대 기계문제를 (M-1)개의 2대 기계문제로 변환하여, 각 문제에 Johnson의 기법을 적용함으로써 얻어지는 (M-1)가지 작업순서 중 최적의 순서를 선택하는 기법을 소개하였다.

(6) Palmer의 발견적 기법 (PALMER)

1965년 Palmer [12]는 M가지의 가공을 요하는 job에 대해 가공시간을 이용하여 기울기 지수를 계산하여, 이를 기준으로 작업순서를 결정하는 이론을 발표하였다.

(7) Bonney와 Gundry의 발견적 기법 (BONNE 1)

1976년 Bonney와 Gundry [13]는 job의 누적된 가공시간들 사이의 기하학적인 관계를 이용하는 기법을 고안하였다. job에 대한 시작과 끝 기울기는, M대의 기계에서 가공의 시작과 끝나는 시간들을 각각 연결하는 2개의 선형회귀선을 추출하여 정의된다. 작업순서는 후속 job의 시작 기울기를 前 job의 끝 기울기에 가급적 부합시킴으로써 결정된다.

(8) Johnson의 기법을 이용한 Bonney와 Gundry의 발견적 기법 (BONNE 2)

1976년 Bonney와 Gundry [13]는 BONNE 1의 적용에서 얻어진 job에 대한 2개의 기울기를 2개의 가상가공시간으로 해석하여 Johnson의 기법을 적용하였다.

(9) King과 Spachis의 발견적 기법 (SPAC 1)

1980년 King과 Spachis [14]가 소개한 5가지 발견적 기법중, 그들이 수행도 분석에 근거하여 2가지 기법(즉, SPAC 1과 SPAC 2)을 본 연구에서 고려한다. 그들은 총 처리시간을 최소화하는 것은 기계의 총 유휴시간을 최소화하는 것과 동일하다는 주장을 근거로 새로운 기

법을 개발하였다. 작업순서는 job과 job사이의 지연을 최소화 시키면서 결정된다.

(10) 가중치를 이용한 King과 Spachis의 발견적 기법 (SPAC 2)

그들은 가공순서상 처음에 소요되는 기계들에서의 지연이 총처리시간에 미치는 영향은 나중에 소요되는 기계들에서의 지연에 비해 극히 미미하다는 사실을 발견하였다. 이에 따라 job과 job사이의 지연에 대한 벌칙으로서, 기계의 가공순서에 따른 가중치의 사용을 SPAC 1 과정에 추가하였다.

(11) Gelders와 Sambandam의 발견적 기법 (GELDER)

1978년 Gelders와 Sambandam [15]은 job의 늦음 (Lateness)과 가공을 위해 job이 기다려야하는 총 시간의 두가지 요인을 이용하여 정의되는 독촉지수 (Dispatching index)에 의한 작업순서 결정기법을 소개하였다.

(12) NAWAZ의 발견적 기법 (NEH)

1983년 NAWAZ 외 3인 [16]에 의해 개발된 기법으로, job의 총가공시간의 크기는 작업순서결정에 중요한 영향을 미친다는 가정에 근거를 두고 있다. 이 기법은 먼저 job의 총 가공시간들이 증가하지 않는 차례로 순서가 부여된 job의 집합으로부터 처음 2개의 job을 가지고 구성할 수 있는 모든 부분순서 (Partial sequences) 중 최소의 총 처리시간을 갖는 최적 부분순서를 결정한다. 다음, 현재의 최적부분순서에서 job의 상대적 위치를 지키면서 집합으로부터 다음 차례의 job을 현재의 최적부분순서의 모든 가능한 위치에 첨가하여 얻어진 새로운 부분순서 가운데서 최적의 부분순서를 선택하는 절차를 반복하여 모든 job의 작업순서를 결정한다.

2.2 발견적 기법의 개발

본 연구에서 제안된 발견적 기법은 이미 소개된 기존의 기법들이 사용한 기본적인 이론을 합리적으로 변형 또는 결합한 것이다.

(1) Petrov 발견적 기법과 Palmer 발견적 기법의 결합(PETRO2)

Petrov의 발견적 기법(PETRO1)에서 원래의 M대의 기계를 가상의 2대 기계문제로 변환시킬 때 정의되는 job 가공시간의 계산에, Palmer의 가중치 개념을 적용한다. N개 job에 대한 2개의 가상 가공시간(t_{1i} 과 t_{2i})을 각각 다음과 같이 계산한다.

$$t_{1i} = \sum_{j=1}^{k_1} \frac{(2j - k_1 - 1)}{2} \cdot t_{ij}, \quad i = 1, 2, \dots, N$$

$$t_{2i} = \sum_{j=k_2}^M \frac{(2j - 3k_1 + 1)}{2} \cdot t_{ij}, \quad M \text{이 홀수인 경우}$$

$$= \sum_{j=k_2}^M \frac{(2j - 3k_1 - 1)}{2} \cdot t_{ij}, \quad M \text{이 짝수인 경우}$$

여기서, M이 홀수인 경우에는 $k_1 = k_2 = [M/2] + 1$, 그리고 M이 짝수인 경우에는 $k_1 = [M/2]$, $k_2 = [M/2] + 1$ 이다.

작업순서 결정을 위해 PETRO1에서와 같이 Johnson의 기법을 적용한다.

(2) Campbell의 3인의 발견적 기법과 Dannenbring 발견적 기법의 결합(CAMP2)

Campbell의 3인의 발견적 기법(CAMP1)에 Dannenbring의 가중치 개념을 도입한다. 따라서 CAMP1의 k번째 소문제에서 job i의 2개의 가상가공시간 (t_{ik} 와 t_{ik}^*)을 다음과 같이 정의한다.

$$t_{ik} = \sum_{j=1}^M (M - j + 1) \cdot t_{ij}, \quad i = 1, 2, \dots, N$$

$$t_{ik}^* = \sum_{j=1}^k (M - j + 1) \cdot t_{i, m-g+1}, \quad k = 1, 2, \dots, M-1$$

각 소문제의 작업순서 결정을 위해 Johnson의 기법을 적용하며, 얻어진 (M-1)가지 순서중 최적의 하나를 최종적으로 선택한다.

(3) Campbell의 3인의 발견적 기법과 Page 발견적 기법의 결합(CAMP3)

Page의 address 지수 [17]개념을 CAMP1에 적용한다. 따라서 CAMP1의 k번째 소문제에서 job i에 대한 address 지수(g^k)

를 다음과 같이 정의한다.

$$g^k = \text{sgn}(t_{ik} - t_{ik}^*) / \min(t_{ik}, t_{ik}^*)$$

작은 address 지수의 job 순서에 따라 작업 순서를 결정한다.

(4) 총 처리시간의 下限値를 이용한 발견적 기법(DELTA)

총 처리시간의 실제값에 근사한 하한치를 합리적으로 계산할 수 있다면, 하한치를 최소화하는 것은 총처리시간을 최소화하는 문제와 同一하다고 할 수 있다.

δ 는 작업순서가 이미 결정된 job들의 집합을 나타내고, $\bar{\delta}$ 는 아직 순서가 결정되지 않은 나머지 job들의 집합을 의미한다고 하자, 또 δ 에 소속되어 있는 job i가 가공순서상 마지막으로 거쳐야 할 기계 m에서의 지연을 D_{im} , 그리고 부분작업순서 δ 에 소속된 job들이 마지막 기계 m에서 가공이 모두 완료되는 시간을 $S(\delta, m)$ 이라 표시하자. 그러면 job i가 현재의 부분 작업순서 δ 에 포함될 때 실제 총 처리시간의 하한치(T_i)를 아래의 식에 의해 계산할 수 있다.

$$T_i = S(\delta, m) + D_{im} + t_{im} + \sum_{\substack{i \in \bar{\delta} \\ i=1}} t_{im}$$

여기서 위의 식을 고찰하여 보면, 지연 D_{im} 을 제외한 나머지 항은 모두 常數임을 알 수 있다. 따라서 마지막 기계에서 가장 짧은 지연을 갖는 job을 현재의 부분작업 순서에 후속 job으로 추가함으로써 총 처리시간의 최소화를 꾀할 수 있다. 이 과정을 모든 job들의 작업순서가 완성될 때까지 반복한다.

3. 動的 flowshop 電算模擬實驗模型

動的 flowshop 문제에서는, 새로운 job들이 일정한 시간에 걸쳐 확률적 분포를 따르며 계속 도착하여 기계의 이용상태에 따라 대기하고 있는 job들에 추가된다. 또 job 가공시간은 확률적 분포를 따른다. 動的 flowshop 문제에서

작업순서 결정을 위한 가장 이상적인 방법은 새로운 job이 도착할 때마다 최적 기법을 적용하여 대기하고 있는 job들의 순서를 결정하는 것이다. 그러나 기법의 빈번한 적용에 의한 작업순서 결정은 계산상이나 이론적인 관점에서 매우 비 실용적이다. 여기서 작업순서 결정에 대한 적당한 빈도수 결정은 매우 중요한 문제라 할 수 있다.

動的 flowshop 문제의 전산 모의 실험을 위해 다음의 조건들을 설정한다.

- (1) 시간 0에 10개의 job들이 첫번째 가공을 받기 위해 대기하고 있다.
- (2) 시간 0에 모든 기계는 이용 가능하다.
- (3) 총 100개의 job이 간격 11과 25시간 단위를 가진 Uniform 분포의 간격을 따르며 작업장에 도착한다.
- (4) Job 가공시간은 각각 평균치 2인 10개의 Exponential 분포의 합으로 결정된 Erlang 분포로부터 추출된 값을 평균치로 하는 Exponential 분포를 따른다.
- (5) 작업순서 결정은 정해진 시간 간격마다 한가지의 선택된 발견적 기법에 의해 반복적으로 이루어 진다.

조건 (4)에서 Erlang 분포 사용에 대한 주요 이유는 Erlang 분포에 따른 분산치가 매우 작다는 특성에 근거를 두고 있다.

動的 flowshop 문제의 discrete event 전산 모의 실험을 위해 사용된 SLAM은 Pegden과 Pritsker [18]에 의해 개발되었다. SLAM은 Network 모델, discrete event 모델, continuous 모델에 모두 가능한 매우 효과적인 전산 모의言語이다. SLAM을 이용한 動的 flowshop 전산 모의 실험 모형은 다음의 5개 event subroutine을 갖는다. 각 subroutine은 FORTRAN을 이용하여 작성된다.

Event JARV ; 작업장에 job 도착 event

Event FMAC : 첫번째 기계에 job 도착 event

Event AMAC : 첫번째 기계를 제외한 다른

기계에 job도착 event

Event ENDM : 어느 기계에서 job 가공 완료 event

Event JSEQ : 발견적 기법의 적용에 의한 작업순서 결정 event

動的 flowshop 전산 모의 실험 모형에 대한 SLAM 프로그램의 기본구조는 그림 1과 같다. SLAM processor는 5개의 event subroutine 외에도 FORTRAN을 이용하여 작성된 2개의 subroutine을 추가로 조정한다. 즉 실험의 초기상태 설정과 최초의 event를 계획하기 위한 INTLC subroutine과 전산 모의 실험 後 특별한 결과들을 인쇄하기 위한 OPUT subroutine이다. INTLC는 전산 모의 실험의 실제 시행 前, 그리고 OPUT는 실험 後에 각각 SLAM processor에 의해 호출된다. Main 프로그램은 storage array 크기를 설정하기 위한 것으로서 조직의 첫번째 프로그램이다.

以上的의 電算模擬實驗者인 본인에 의해 작성된 프로그램들 외에 SLAM 프로그램은 몇가지 固有 프로그램들을 포함하고 있다. SLAM Reports는 SLAM 固有의 보고형식을 위한 프로그램으로서 변수들과 file들에 대한 統計를 수집 계산한다. 또한 SLAM library는 전산모의실험을 행하는데 필요한 여러가지의 sub 프로그램들로 구성되어 있는데, 특히 확률분포에 따른 난수 산출, event의 진행, 필요한 통계 수집등을 위한 sub 프로그램을 포함하고 있다.

4. 비교평가 및 분석

4.1 Steel의 rank sum 테스트

전산모의실험에 의한 발견적 기법의 유효성을 비교평가하기 전에 여러 다른 기법들의 적용이 총 처리시간에 실질적인 차이를 초래하는지를 Steel의 rank sum 테스트 [19]를 이용하여 규명하였다.

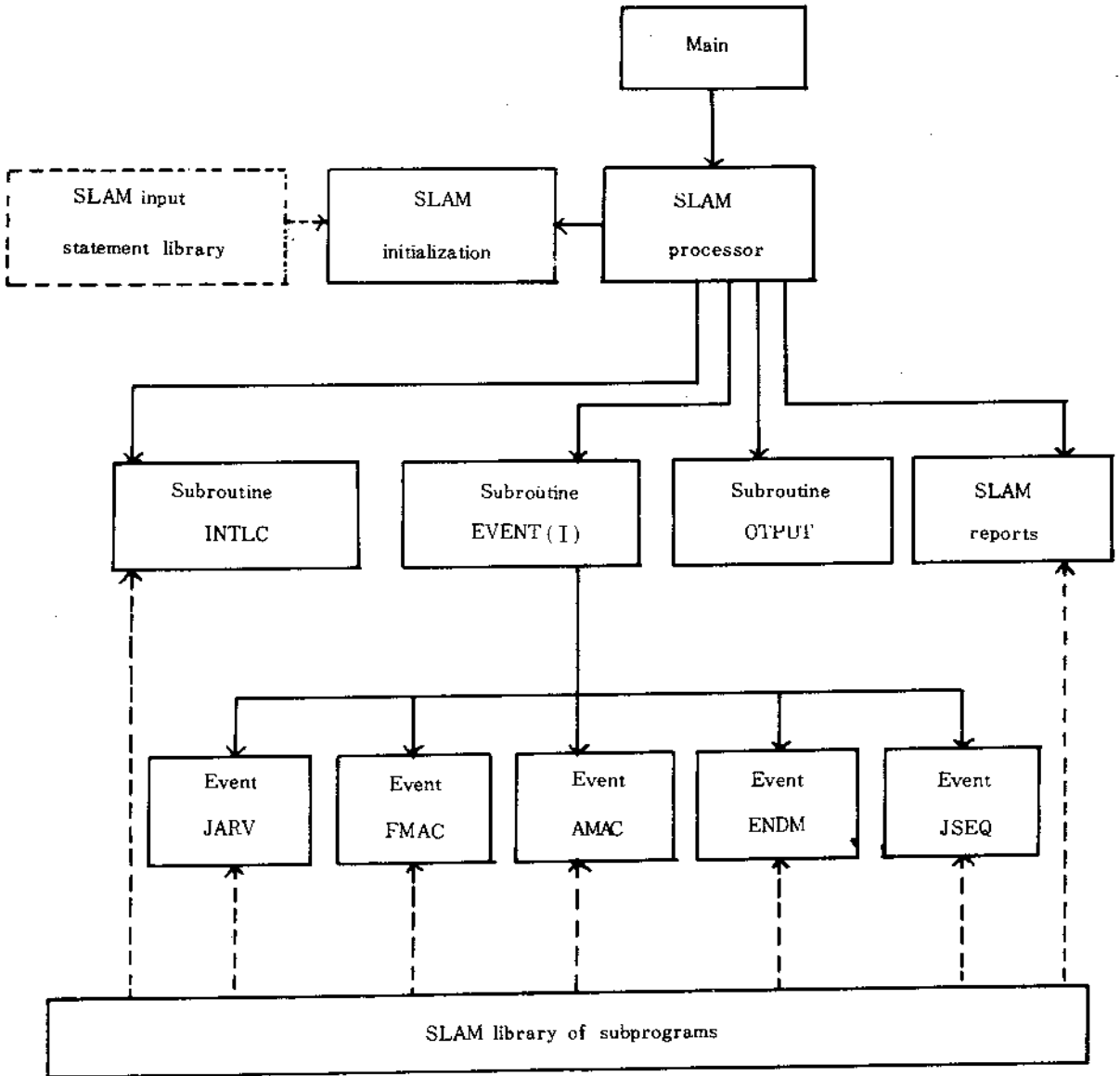


Figure 1. The Organization of SLAM Program for the Dynamic Flowshop Simulation Model

Steel의 테스트를 뒷받침하기 위해 χ^2 goodness-of-fit 테스트를 5% 예리수준에서 시행하였다. 그 결과 발견적 기법들의 적용에 의해 얻어진 총 처리시간의 분포는 대략 정규분포를 따른다는 가정이 입증되었다. Steel의 테스트는 여러 크기의 총 1500가지의 靜的 flowshop 模擬問題를 대상으로 5% 예리율로써 ($\alpha=0.05$)시행하였다.

표2의 결과에 의하면 3-job 문제를 제외한 모든 문제에서 여러 다른 발견적 기법들의 적용은 총 처리시간의 차이에 5%예리율로써 중대한 영향을 미친다. 특히 PETRO2, BONNE1가 최대의 rank sum 통계치(R)를 얻었다는 사실은 靜的 flowshop 문제에서 이들 2가지의 발견적 기법들의 유효성이 매우 낮을 가능성을 암시해 준다. 또 R은 문제의 크

Table 2. Steel's Rank Sum Test for the Static Flowshop Problems ($\alpha=0.05$)

Problem size (N×M)	Sample size	R(i*)+	H ₀
3 × 4	50	484 (2)	A
8	50	489 (10)	A
12	50	484 (10)	A
16	50	490 (10)	A
20	50	487 (10)	A
6 × 4	50	548 (2)	R
8	50	540 (2)	R
12	50	540 (10)	R
16	50	542 (10)	R
20	50	570 (2)	R
9 × 4	50	566 (2)	R
8	50	574 (2)	R
12	50	577 (2)	R
16	50	588 (2)	R
20	50	602 (10)	R
15 × 4	20	593 (2)	R
8	20	610 (2)	R
12	20	603 (2)	R
16	20	605 (2)	R
20	20	610 (2)	R
20 × 4	20	597 (2)	R
8	20	610 (2)	R
12	20	610 (2)	R
16	20	610 (2)	R
20	20	610 (2)	R
30 × 4	20	608 (2)	R
8	20	610 (2)	R
12	20	610 (2)	R
16	20	610 (2)	R
20	20	610 (2)	R
H ₀ : $\mu_1 = \mu_2 \dots = \mu_{16}$			A Accepted
H ₁ : $\mu_1 \neq \mu_2 \dots \neq \mu_{16}$			R Rejected

+R denotes the maximum rank sum and i* denotes the heuristic number (see Table 1) for which the maximum rank sum was achieved.

기가 커짐에 따라 증가하고 있는데 이것은 기법들 간의 총 처리시간의 차이는 문제가 크기가 커짐에 따라 증가함을 의미한다.

3-job 문제들에서 총 처리시간의 차이가 거의 나타나지 않은 이유는 다음과 같이 설명될 수 있다. 첫째, 3-job 문제에서는 단지 6가지의 가능한 작업 순서를 갖고 있기 때문에 최적해를 얻을 수 있는 확률이 매우 높다. 둘째, 작은 분산치를 갖는 Erlang 분포로부터 job 가공시간이 결정되므로 기법의 유효성에 대해 random error의 영향이 극히 작다.

4.2 전산 모의 실험 결과의 분석

전산모의 실험 결과에 의한 발전적 기법들의 비교평가를 위해 Bechhofer의 Multiple Ranking Procedure (MRP) [19]를 사용하였으며, 정확한 분석을 위해 총 100번의 반복 실험을 실시하였다. 표 3은 靜的 그리고 動的

flowshop 문제에서 각 발전적 기법들에 의해 얻어진 평균 총 처리시간의 추정 에러율과 16가지 기법들의 평균등급을 보여준다.

먼저, 靜的과 動的 flowshop 문제에서 기법의 유효성에 대한 등급이 NEH를 제외하고는 서로 일치하지 않음을 알 수 있다. 표 3에 기재된 靜的문제에서의 유효성 등급은 1984년 본인이 발표한 연구논문[20]에 의거한 것이다. NEH는 靜的과 動的 문제에서 모두 유효성이 가장 뛰어나다. SPAC1, PETRO2, GUPTA, RAP, PALMER, BONNE2는 靜的 문제에서 보다 動的 문제에서 더 효과적임을 보여준다. 이러한 변화는 動的 문제에 대한 이들 기법 고유의 유연성에 기인된 것이라 사려된다. 반면에 CAMP1을 비롯하여 PETRO1, SPAC2, GELDER는 靜的문제에서 더 효과적이다.

Dynamic flowshop 문제에 대한 평균 총 처리

Table 3. The Common Comparison Test and Bechhofer et al.'s MRP for Flowshop Problems

Model	Evaluatory measures	Frequency	Heuristic No.															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Dynamic flowshop model	Estimated error rate of mean makespan (%)	40	1.95	8.58	1.26	1.53	1.08	1.65	0.63	0.9	0.0	8.16	4.53	0.63	7.5	9.51	2.73	9.15
		80	2.07	8.49	1.41	1.5	0.63	1.77	0.45	0.72	0.0	7.83	4.05	0.3	6.21	8.91	0.63	8.61
		120	2.22	6.39	1.59	1.5	1.44	1.56	0.6	0.6	1.02	6.72	4.44	0.45	5.7	8.22	0.0	10.65
		160	2.94	5.73	0.84	1.44	1.2	1.32	0.9	0.96	1.35	5.94	3.75	0.69	3.93	7.44	0.0	8.91
		200	2.82	5.34	1.29	1.47	1.41	1.38	1.11	1.26	1.71	5.61	3.57	0.93	3.93	6.54	0.0	8.4
		240	3.39	4.86	2.4	1.35	2.01	2.01	1.74	2.01	2.61	5.73	3.84	1.2	3.15	6.39	0.0	7.41
	280	3.09	3.9	2.1	1.11	1.59	1.59	1.89	1.23	2.19	4.98	2.34	1.86	2.91	5.4	0.0	3.15	
	Mean error rate (%)		2.64	6.18	1.56	1.41	1.33	1.61	1.05	1.1	1.27	6.4	3.79	0.87	4.71	7.49	0.48	8.04
	Effective ranking		10	13	8	7	6	9	3	4	5	14	11	2	12	15	1	16
Static flowshop model	Mean error rate (%)		3.04	9.82	4.79	3.13	2.57	1.9	1.42	1.79	2.97	7.22	5.21	2.58	3.37	6.4	0.0	6.88
	Effective ranking		8	16	11	9	5	4	2	3	7	15	12	6	10	13	1	14
	Average Computation time (0.001 sec)		1.52	4.71	0.53	1.05	4.87	5.49	5.61	5.83	0.68	0.6	0.63	73.61106.6211.12	16.53	7.1		

시간의 추정 에러율로부터 기법들의 유효성이 작업순서 결정빈도에 따라 변화한다는 사실을 발견할 수 있다. 이것은 일단 결정된 작업 순서는 다음 결정때까지의 기간동안만 유효하다는 사실로 설명될 수 있다. 즉, 40이나 80시간 단위의 높은 빈도에서는 결정된 작업 순서는 처음 몇개의 job 들에게만 적용이 된다. 반면에 200, 240, 280시간 단위의 낮은 빈도에서는 다음 작업순서 결정때까지 순서를 결정받지 못한 job이 존재되어, 이들은 작업순서에 포함된 모든 job들의 가공이 끝난 후 first-in first-out 규칙에 의해 가공을 받는다. 40과 80의 높은 빈도에서는 PALMER가 가장 뛰어나며, 나머지 빈도에서는 NEH가 가장 효과적이다. 전산 모의 실험결과 40시간 단위의 결정빈도에 의한 PALMER의 적용이 가장 짧은 총 처리시간을 산출하였다.

여기서 우리는 動的 flowshop 작업순서 결정문제에서는 가장 적합한 기법의 선택과 더불어 결정빈도의 합리적 결정이 매우 중요하고도 어려운 문제임을 인식할 수 있다.

표 4는 動的 flowshop 전산 모의 실험에 의해 얻어진 두 가지의 다른 유효성 측정치, 즉 평균재고량과 평균기계이용율에 관한 통계치를 보여준다. 괄호속의 수치는 평균기계 이용율을 나타낸다. 평균재고량의 정보는 buffer area의 설계에 영향을 미치며 기계 이용율은 기계의 부하에 대한 중요한 사전 정보를 제공하여 준다.

끝으로, 전산 모의 실험 과정에서 job 가공 시간에 대한 분산이 증가함에 따라 발전적 기법들의 유효성의 차이는 감소한다는 사실을 확인하였다. 이것은 1977년 Dannenbring [10]의 data error에 관한 실험 결과와 일치한다.

Table 4. Average In-process Inventory and Average Machine Utilization Generated by Running the Dynamic Flowshop Simulation Model

Heuristic No. Frequency	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
40	24.60 (0.86)	24.26 (0.86)	24.87 (0.86)	24.28 (0.84)	24.93 (0.86)	24.63 (0.86)	24.76 (0.86)	24.26 (0.86)	24.25 (0.87)	22.50 (0.84)	25.01 (0.85)	24.90 (0.86)	22.63 (0.86)	24.03 (0.84)	23.85 (0.86)	22.36 (0.84)
80	24.58 (0.86)	24.27 (0.86)	24.82 (0.86)	24.14 (0.83)	24.79 (0.86)	24.78 (0.86)	24.73 (0.86)	24.22 (0.86)	24.16 (0.86)	22.81 (0.84)	24.93 (0.85)	24.64 (0.86)	22.80 (0.86)	24.02 (0.84)	23.70 (0.86)	22.33 (0.84)
120	24.54 (0.86)	24.27 (0.86)	24.72 (0.86)	24.01 (0.83)	24.76 (0.86)	24.57 (0.86)	24.56 (0.86)	24.22 (0.86)	24.29 (0.86)	23.06 (0.84)	24.93 (0.85)	24.60 (0.86)	22.90 (0.86)	24.03 (0.84)	23.68 (0.86)	22.64 (0.83)
160	24.38 (0.85)	24.27 (0.86)	24.63 (0.86)	24.01 (0.83)	24.56 (0.86)	24.48 (0.87)	24.55 (0.86)	24.14 (0.86)	24.23 (0.86)	23.24 (0.84)	24.65 (0.85)	24.50 (0.86)	23.77 (0.86)	24.01 (0.84)	23.74 (0.86)	22.90 (0.84)
200	24.30 (0.85)	24.28 (0.85)	24.54 (0.85)	23.97 (0.84)	24.37 (0.85)	24.30 (0.85)	24.42 (0.85)	24.17 (0.85)	24.29 (0.85)	23.55 (0.84)	24.46 (0.85)	24.50 (0.85)	23.77 (0.85)	24.01 (0.84)	23.80 (0.86)	23.10 (0.84)
240	24.24 (0.85)	24.28 (0.85)	24.45 (0.85)	23.97 (0.84)	24.18 (0.85)	24.14 (0.85)	24.21 (0.85)	24.12 (0.85)	24.25 (0.85)	23.67 (0.84)	24.32 (0.85)	24.33 (0.85)	23.89 (0.85)	24.01 (0.84)	23.70 (0.86)	23.38 (0.84)
280	24.21 (0.85)	24.28 (0.85)	24.31 (0.85)	23.90 (0.84)	24.11 (0.85)	24.07 (0.85)	24.05 (0.85)	24.06 (0.85)	24.22 (0.85)	23.87 (0.84)	24.22 (0.85)	24.20 (0.85)	23.90 (0.85)	24.01 (0.84)	23.78 (0.85)	23.64 (0.84)

5. 결 론

Job이 일정한 시간 동안 확률적 분포를 따르면서 계속해서 작업장에 도착하여 가공을 위해 기계의 이용상태에 따라 대기하고, 또 job 가공 시간이 확률적 분포를 따르는 flowshop을 動的 flowshop이라 정의한다. 본 연구에서는 靜的 flowshop 문제의 작업 순서 결정을 위해 개발된 총 16가지의 발견적 기법을 소개하고 이들 기법들의 動的 flowshop 문제에서의 유효성을 SLAM을 이용한 전산 모의 실험을 통하여 비교 평가하였다. 얻어진 결론을 정리하면 다음과 같다.

(1) 발견적 기법들의 적용에 의해 얻어진 작업순서는 5%에러율로써 총 처리시간의 차이에 중대한 영향을 미친다.

(2) Job 가공시간에 대한 분산이 증가함에 따라 발견적 기법의 적용에 의한 총 처리시간의 차이는 감소한다.

(3) 작업순서 결정빈도는 발견적 기법의 유효성에 영향을 미친다.

(4) 動的 flowshop 문제의 전산 모의 실험은 buffer area 크기와 기계의 부하에 관한 중요한 정보를 제공한다.

(5) 총 처리시간을 최소화 하기 위한 일반적 인 動的 flowshop 문제에서 NEH가 16가지 발견적 기법중 가장 효과적이며, SPAC1, CAMP 2, 그리고 CAMP 3 순으로 이어진다. 특히, 높은 빈도의 작업순서 결정에서 PALMER는 매우 효과적이다.

References

1. Baker, K. R., *Introduction to Sequencing and Scheduling*, John Wiley and Sons, Inc., New York, pp. 136-137, 1974.
2. Little, J., Murty, K., Sweeney, D., and Karel, C., "An Algorithm for the Travelling Salesman Problem," *Operational Research Quarterly*, Vol. 11, No. 6, 1963.
3. Lomnicki, Z., "A Branch and Bound Algorithm for the Exact Solution of the Three Machine Scheduling Problem," *Operational Research Quarterly*, Vol. 16, No. 1, 1964.
4. Bestwick, P. F., and Hastings, T. A. J., "A New Bound for Machine Scheduling Problem," *Operational Research Quarterly*, Vol. 27, No. 2, 1976.
5. Littger, K. W., "A New Approach for the Optimum Solution of the M by J Production Scheduling Problem," *Int. J. Prod. Res.*, Vol. 14, No. 1, 1976.
6. Baker, K. R., and Schrage, L. E., "Finding an Optimal Sequence by Dynamic Programming," *Operations Research*, Vol. 26, No. 1, 1978.
7. Johnson, S., "Optimal Two and Three Stage Production Schedules with Setup Times Included," *Nav. Res. Log. Q.*, Vol. 1, 1954.
8. Petrov, V. A., *Flow Line Group Production Planning*, Business Publication Ltd., London, 1966.
9. Gupta, J. N. D., "A Functional Heuristic Algorithm for the Flowshop Scheduling Problem," *Operational Research Quarterly*, Vol. 22, No. 1, 1971.
10. Dannenbring, D. G., "An Evaluation of Flow-Shop Sequencing Heuristics," *Management Science*, Vol. 23, No. 11, 1977.
11. Campbell, H. G., Dudek, R. A., and Smith, M. L., "An Heuristic Algorithm for the n Job m Machine Sequencing Problem," *Management Science*, Vol. 16, No. 10, 1970.

12. Palmer, D. S., "Sequencing Jobs through a Multi-Stage Process in the Minimum Total Time-a Quick Method of Obtaining a Near Optimum," *Operational Research Quarterly*, Vol. 16, No. 1, 1965.
13. Bonney, M. C., and Gundry, S. W., "Solutions to the Constrained Flowshop Sequencing Problem," *Operational Research Quarterly*, Vol. 27, No. 4, 1976.
14. King, J. R., and Spachis, A. S., "Heuristics for Flow-Shop Scheduling," *Int. J. Prod. Res.*, Vol. 19, No. 3, 1980.
15. Gelders, L. F., and Sambandam, N., "Four Simple Heuristics for Scheduling a Flowshop," *Int. J. Prod. Res.*, Vol. 16, No. 3, 1978.
16. Nawaz, M., Ensore, E., and Ham, I., "A Heuristic Algorithm for the m Machine n Job Flow-Shop Sequencing Problem," *OMEGA*, Vol. 11, No. 1, 1983.
17. Page, E. S., "An Approach to Scheduling Jobs on Machines," *J. Royal Statistical Society, Series B*, Vol. 23, 1961.
18. Kleijnen, J. P. C., *Statistical Techniques in Simulation*, Part II, Dekker, New York, 1975.
19. Park, Y. B., Pegden, C. D., and Ensore, E. E., "A Survey and Evaluation of Static Flowshop Scheduling Heuristics," *Int. J. Prod. Res.*, Vol. 22, No. 1, pp. 127-141, 1984.