

## n/m/D/F/F<sub>max</sub> 스케줄링 문제의 휴리스틱 알고리듬(Ⅱ) —A Development of Heuristic Algorithms for the n/m/D/F/F<sub>max</sub> Scheduling Problem—

崔 聖 雲 \*  
魯 仁 珪 \*\*

### Abstract

This paper considers a multi-stage flowshop scheduling problem where the setup times of jobs depend on immediately preceding jobs. Three heuristics algorithms, CAMPBELL, PEIDAN and CAMRING are proposed. The performance measure is a minimization of makespan. The parameters of simulation model are PS(ratio of the processing times to setup times), M(number of machines), and N(number of job). This simulation model for each algorithm is a  $4 \times 3 \times 3$  factorial design with 360 observations. The makespan of the proposed heuristic algorithms is compared with the optimal makespan obtained by the complete enumeration of schedules. This yardstick of comparison is defined as a relative error. The mean relative error of CAMPBELL, PEIDAN, and CAMRING algorithms are 4.353%, 7.908%, and 8.578%, respectively.

The SPSS is used to analyse empirical results. The experimental results show that the three factors are statistically significant at 5% level.

### 1. 서 론

n개 작업, m대 기계인 훌로우·샵(Flow Shop) 스케줄링 문제는 준비시간이 작업순서에 독립적인 경우를 중심으로 알고리듬이 많이 연구되었다[1-17]. 즉 가공시간이 준비시간에 포함되는 것으로 가정을 하였으나 실제 현장에서는 준비시간이 작업순서에 큰 영향을 받기 때문에, 이것들을 분리하여 준비시간이 작업순서에 종속적인 문제로 다루어져야 한다.

n개 작업 m대 기계인 훌로우·샵 SDST(Sequence Dependent Setup Time) 스케줄링 문제에 대한 연구로는 2대 기계인 경우[18, 19, 20] 및 m대 기계인 경우[21] 등이 있으나, 이중 m대 기계인 경우는 계산량이 과중한 MILP(Mixed Integer Linear Programming)로 문제를 해결하였다. Gupta[22]는 n/m/D/J/F<sub>max</sub> 스케줄링 문제에서의 분기·한계(Branch and Bound) 기법을 연구하였으며, Gupta[23]는 제한된 재공품 보관 장소가 있는 경우의 n/m/D/F/F<sub>max</sub> 스케줄링 문제에서의 휴리스틱 알고리듬을 연구하였다. 그러나 현장에서도 쉽게 사용할 수 있으며 최적해에

비하여 그다지 효율이 떨어지지 않는 n/m/D/F/F<sub>max</sub> 스케줄링 문제에서의 휴리스틱 알고리듬을 연구할 필요가 있다. 따라서 그동안 많은 연구가 이루어졌던 n/m/I/F/F<sub>max</sub> 스케줄링 문제에서의 휴리스틱 알고리듬을 기초로 하여 본 논문에서는 n/m/D/F/F<sub>max</sub>에서 적용할 수 있는 새로운 3가지의 휴리스틱 알고리듬을 개발하기로 한다.

1954년 Johnson[1]은 대부분의 연구에서 중요한 토픽이 될 수 있는 2대 기계인 훌로우·샵 스케줄링 문제에 대한 알고리듬을 개발하였으며 1966년 Petrov[7]는 이러한 Johnson 알고리듬을 기초로 m대 기계로 확장하였다.

1965년 Palmer[5]은 m대 기계의 작업에 대한 경사지수(Slope Index)를 사용하여 계산된 지수값의 내림차순으로 작업순서를 정하였다.

1970년 Campbell et al.[23]은 기계가 m대일 때( $m-1$ )개 까지 2대 기계 문제로 생각하여 각각에 대해 Johnson 알고리듬을 적용하여 문제를 해결하였으며, 1977년 Dannenbring[32]은 경사지수와 Campbell et al. 알고리듬과 유사한 가공차를 가공시간에 고려하여 Johnson 알고리듬으로 문제를 해결하였다.

따라서 본 논문에서는 이러한 문헌 조사와 통하여 새로운 휴리스틱 알고리듬을 개발하여 완전열거법에 의한 최적해와의 상대오차(Relative Error)를 비교·

\*경원대학교 산업공학과 전임강사

\*\*한양대학교 산업공학과 교수

접수 : 1989. 4. 29.

평가(4~8%)하였다. 또한 가공시간과 준비시간의 비율, 기계의 수, 작업의 수 등의 파라미터를 변화시킬 때 따르는 상대오차의 변화를 SPSS 통계 패키지[24]로 분석하여 보았다.

## 2. n/m/D/F/Fmax 스케줄링 문제

### 2·1 문제의 가정

본 논문에서 연구하고자 하는 문제는 준비시간이 가공시간에 종속적인 경우 홀로우·샵 스케줄링에서 생산기간을 최소화하는 문제로서 다음의 가정을 전제로 한다.

- (1) 기계 수와 형태는 알고 있고 변하지 않는다.
- (2) 첫번째 작업이 초기에 가공되는 시점에서 모든 기계는 이용 가능하다.
- (3) 모든 기계는 항상 이용 가능하다. 즉 고장, 보수, 교대 작업, 일간 변동에 따른 변화가 없다.
- (4) 각 작업은 4가지 상태 즉 다음 기계에 대한 정체 상태, 준비상태, 가공상태, 가공 마무리상태 중 하나이다.
- (5) 어떠한 작업도 완료하기 전에는 시스템을 떠날 수 없다. 어떤 경우에도 작업분할, 타부서 이관, 또는 외부하청을 고려할 수 없다.
- (6) 한 기계가 특정 작업을 가공하여 끝날 때까지는 어떤 다른 작업도 가공될 수 없다.
- (7) 모든 작업에 대한 기계의 가공순서는 같으며 항상 고정되어 있다.
- (8) 각 기계에 대한 작업의 준비시간은 가공순서에 종속적이다.
- (9) 다음 기계로의 작업의 이동시간은 무시해도 좋다.
- (10) 모든 작업의 우선 순위는 같다.
- (11) 모든 작업은 초기에 가공준비가 되어 있다.
- (12) 각 작업에 대한 남기는 고려되지 않는다.

### 2·2 기호

N : 작업

M : 기계

n : 작업의 수

m : 기계의 수

a : 보조 문제의 수 ( $a=m-1$ )

j : 작업번호 ( $j=1, 2, \dots, n$ )

i : 작업의 전단계 작업번호 ( $i=1, 2, \dots, n$ )

l : 보조 문제 번호 ( $l=1, 2, \dots, a$ )

k : 기계 ( $k=1, 2, \dots, m$ )

p : 가공시간

r : 작업순서 위치

S : 준비시간

$T^l_{rj1}$  : 첫번째 가상 기계에서의 가공 시간과 준비 시간의 합.

$T^l_{rj2}$  : 두번째 가상 기계에서의 가공시간과 준비 시간의 합.

A :  $T^l_{rj2} \geq T^l_{rj1}$ 을 초과하는 j의 집합

B :  $T^l_{rj1} \geq T^l_{rj2}$ 을 초과하는 j의 집합

## 3. 휴리스틱 알고리듬의 개발

### 3·1 CAMPBEL 알고리듬

CAMPBEL(Campbell et al[8]) 알고리듬은 ( $m-1$ )개의 보조 문제에 Johnson 알고리듬을 적용하여 이는 각 끝 기계를 변동부분으로 놓고 중심 기계를 고정부분으로하여 생산기간을 최소화한다.

CAMPBEL 알고리듬에 대한 절차는 다음과 같다.

단계1. 가공시간의  $n \times m$  행렬 P를 설정한다.  $P_{jk}$ 는  $k=1, 2, \dots, m$ 번째 기계에 대한  $j=1, 2, \dots, n$ 번째 작업의 가공시간이다.

단계2. 준비시간의  $n \times m$  행렬 S를 설정한다.  $S_{ijk}$ 는  $k$ 번째 기계에서 작업  $j=(1, 2, \dots, n)$ 가 작업  $j$ 로 변환 경우의 준비시간이다. 단  $i=j$ 면 모든  $k$ 에 대해서  $S_{ijk}=0$ 이다.

단계3. “2대의 기계” 문제로 변형된 보조 문제의 수  $a$ 는 ( $m-1$ )개까지 고려한다.

단계4. 첫번째 보조 문제는  $l=1$ 로 놓는다.

단계5.  $r=(1, 2, \dots, n)$ 은 작업순서의 위치를 나타내며 ( $r$ )은 배정된 작업의 위치이다. 작업이 배정된 첫번째 위치는  $r=1$ 로 놓는다.

단계6. 모든 작업에 대해서

$$T^l_{rj1} = \sum_{k=1}^m [P_{jk} + S_{(r-1)jk}] \text{와}$$

$$T^l_{rj2} = \sum_{k=m+1}^n [P_{jk} + S_{(r-1)jk}] \text{를 계산한다.}$$

단계7.  $A = \{i \mid T^l_{rj1} < T^l_{rj2}\}$ 로 놓는다. 가장 작은  $T^l_{rj1}$ 을 갖는 작업  $j^*$ 를 r위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. A가 공집합이면 다음 단계로 가고 그렇지 않으면 단계 9로 간다.

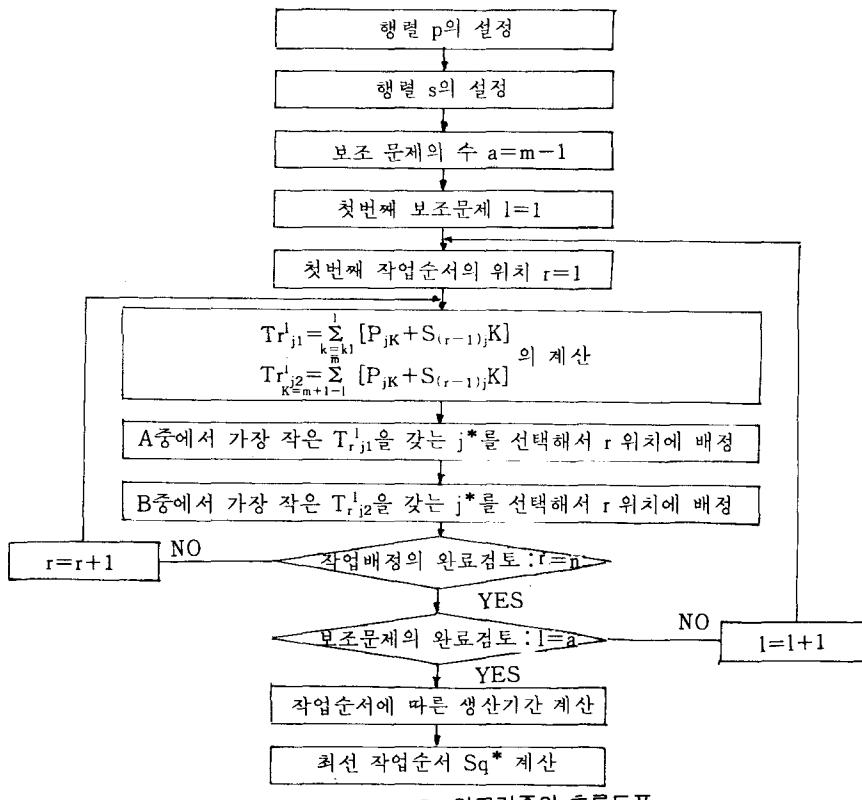
단계8.  $B = \{j \mid T^l_{rj1} > T^l_{rj2}\}$ 로 놓는다. 가장 큰  $T^l_{rj2}$ 를 갖는 작업  $j^*$ 를 r위치에 배정한다.

단계9.  $r < n$ 이면  $r=r+1$ 로 놓고 단계6으로 가고  $r=n$ 이면 다음 단계로 간다.

단계10.  $l > a$ 이면  $l=l+1$ 로 놓고 단계5로 가고  $l=a$ 이면 다음 단계로 간다.

단계11. 원래의  $n \times m$  가공시간 행렬과  $n \times m$  준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계12. 가장 최소인 생산기간을 갖는 작업순서  $Sq^*$ 를 선택한다.



이러한 CAMPBEL 알고리듬의 흐름도표는 그림 1과 같다.

### 3·2 PEIDAN 알고리듬

PEIDAN(Petrov[7] and Inverse Dannenbring[12]) 알고리듬은 Johnson알고리듬을 확장한 Petrov알고리듬에 Dannenbring알고리듬의 지수를 역으로 가중하여 개발하였다.

PEIDAN알고리듬에 대한 절차는 다음과 같다.

단계1. 가공시간의  $n \times m$  행렬 P를 설정한다.  $P_{jk}$ 는  $k (=1, 2, \dots, m)$ 번째 기계에 대한  $J (=1, 2, \dots, n)$ 번째 작업의 가공시간이다.

단계2. 준비시간의  $n \times m$  행렬 S를 설정한다.  $S_{ijk}$ 는  $k$ 번째 기계에서 작업  $i (=1, 2, \dots, n)$ 가 작업  $j$ 로 변할 경우의 준비시간이다. 단  $i=j$ 이면 모든  $k$ 에 대해서  $S_{ijk}=0$ 이다.

단계3.  $r (=1, 2, \dots, n)$ 는 작업순서의 위치를 나타내며 ( $r$ )은 배정된 작업의 위치이다. 작업이 배정된 첫번째 위치는  $r=1$ 로 놓는다.

단계4. 모든 작업에 대해서

$$Tr_{j1} = \sum_{k=1}^{k_1} k \cdot [P_{jk} + S_{(r-1)jk}] 와$$

$$Tr_{j2} = \sum_{k=k_2+1-l}^m k \cdot [P_{jk} + S_{(r-1)jk}] 를 계산한다.$$

단  $m$ 이 짜수이면  $k_1 = \frac{m}{2}$ ,  $k_2 = \frac{m}{2} + 1$ 이며,  $m$ 이 홀수이면  $k_1 = k_2 = \frac{m-1}{2} + 1$ 이다.

단계5.  $A = |j | T_{rk1} < T_{rj2}|$ 로 놓는다. 가장 작은  $T_{rj1}$ 을 갖는 작업  $j^*$ 를  $r$ 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. A가 공집합이면 다음 단계로 가고 그렇지 않으면 단계7로 간다.

단계6.  $B = |j | T_{rj1} > T_{rj2}|$ 로 놓는다. 가장 큰  $T_{rj2}$ 을 갖는 작업  $j^*$ 를  $r$ 위치에 배정한다.

단계7.  $r < n$ 이면  $r=r+1$ 로 놓고 단계4로 가고  $r=n$ 이면 다음 단계로 간다.

단계8. 원래의  $n \times m$  가공시간 행렬과  $n \times m$  준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계9. 가장 최소인 생산기간을 갖는 작업순서  $Sq^*$ 를 선택한다.

이러한 PEIDAN알고리듬의 흐름도표는 그림 2와 같다.

### 3·3 CAMPRING 알고리듬

CAMPRING(Campbell et al. and Dannenbring) 알고리듬은  $(m-1)$ 개의 보조 문제로서 Johnson알고리듬을 적용하는 Campbell et al. 알고리듬에 Dannenbring 알고리듬의 경사지수를 가중하여 개발하였다.

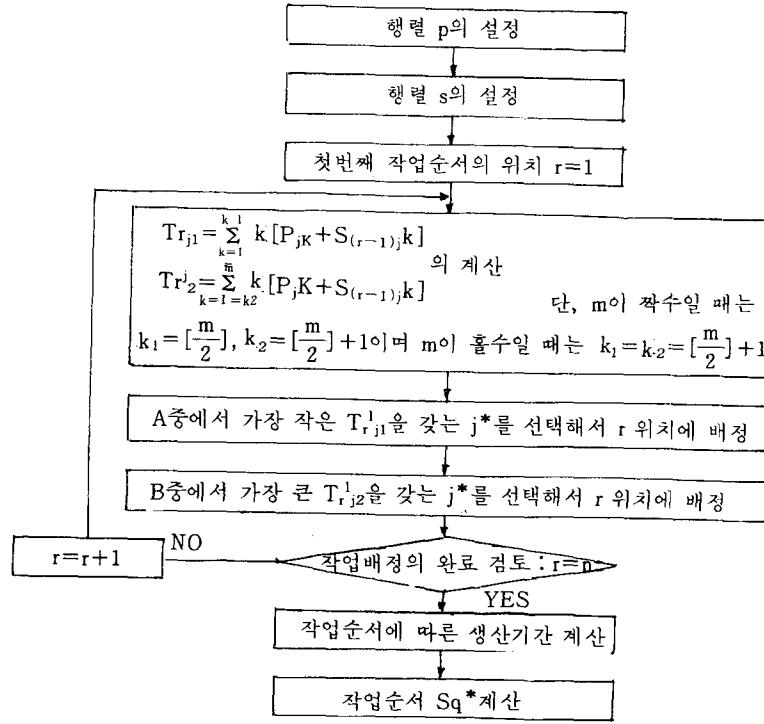


그림 2. PEIDAN 알고리즘의 흐름도표

CAMPRING 알고리듬에 대한 절차는 다음과 같다.

단계1. 가공시간의  $n \times m$  행렬 P를 설정한다.  $P_{jk}$ 는  $k (= 1, 2, \dots, m)$  번째 기계에 대한  $j (= 1, 2, \dots, n)$  번째 작업의 가공시간이다.

단계2. 준비시간의  $n \times m$  행렬 S를 설정한다.  $S_{ijk}$ 는  $k$  번째 기계에서 작업  $i (= 1, 2, \dots, n)$  가 작업  $j$ 로 변환 경우의 준비시간이다. 단  $i=j$  이면 모든  $k$ 에 대해서  $S_{ijk}=0$  이다.

단계3. “2대의 기계” 문제로 변형된 보조 문제의 수  $a$ 는  $(m-1)$  개까지 고려한다.

단계4. 첫 번째 보조 문제는  $l=1$ 로 놓는다.

단계5.  $r (= 1, 2, \dots, n)$  는 작업순서의 위치를 나타내며  $(r)$ 은 배정된 작업의 위치이다. 작업이 배정된 첫 번째 위치는  $r=1$ 로 놓는다.

단계6. 모든 작업에 대해서

$$T^l_{rj1} = \sum_{k=1}^l (m-k+1)[P_{jk} + S_{(r-1)jk}] \text{ 와}$$

$$T^l_{rj2} = \sum_{k=m+1-l}^m (m-k+1)[P_{jk} + S_{(r-1)jk}] \text{ 를 계산한다.}$$

단계7.  $A = \{j \mid T^l_{rj1} < T^l_{rj2}\}$  로 놓는다. 가장 작은  $T^l_{rj1}$  을 갖는 작업  $j^*$  를  $r$  위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다.  $A$  가 공집합이면 다음 단계로 가고 그렇지 않으면 단계9로 간다.

단계8.  $B = \{j \mid T^l_{rj1} > T^l_{rj2}\}$  로 놓는다. 가장 큰

$T^l_{rj2}$  을 갖는 작업  $j^*$  를  $r$  위치에 배정한다.

단계9.  $r < n$  이면  $r=r+1$  로 놓고 단계6으로 가고  $r=n$  이면 다음 단계로 간다.

단계10.  $l < a$  이면  $l=l+1$  로 놓고 단계5로 가고  $l=a$  이면 다음 단계로 간다.

단계11. 원래의  $n \times m$  가공시간 행렬과  $n \times m$  준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계12. 가장 최소인 생산기간을 갖는 작업순서  $Sq^*$  를 선택한다.

이러한 CAMPRING 알고리듬의 흐름도표는 그림 3과 같다.

### 3 · 4 알고리듬의 기본 구조

이러한 3 알고리듬은 결국 (1) Johnson 알고리듬을 1회 적용하느냐 여러번 적용하느냐 (2) 지수의 사용을 어떻게 하느냐에 따라 기본 구조가 형성되어 생성될 때의 수에 영향을 주며 표1과 같다.

표1에서 보면 CAMPBEL 알고리듬과 CAMRING 알고리듬은 Johnson 알고리듬을  $(m-1)$  번 적용하여  $(m-1)$  개의 해를 생성하여 그 중에서 가장 작은 총 처리 시간을 갖는 작업순서를 결정한다. 또한 CAMPBEL 알고리듬에서는 지수를 사용하지 않았으며 PEIDAN 알고리듬과 CAMPING 알고리듬은 지수를 사용하였

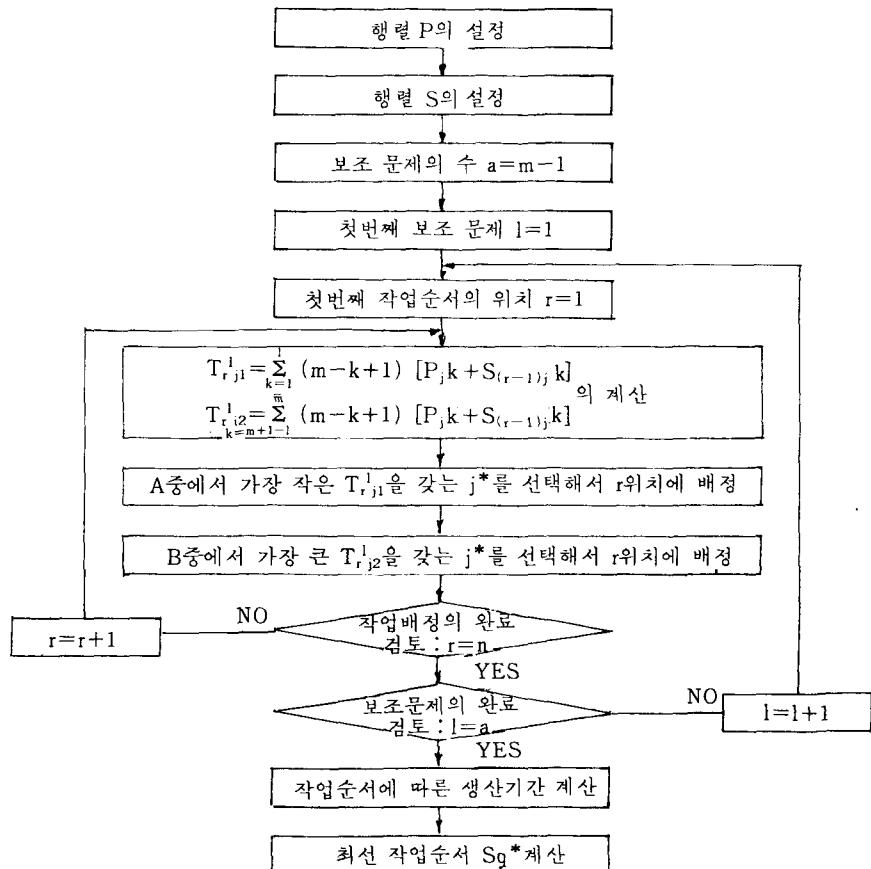


그림 3. CAMPRING 알고리즘의 흐름도표

표 1. 각 휴리스틱 알고리듬의 기본 구조

알고리듬	Johnson 알고리듬		지수	생성된 해의 수(개)
	1회	여러번		
CAMPBEL	—	×	—	m-1
PEIDAN	×	—	×	1
CAMRING	—	×	×	m-
OPTIMAL				n !

\*OPTIMAL : 최적 알고리듬

다. 이 기본 구조는 나중에 최적해와의 상대오차를 평가할 경우 반드시 고려해야 하는 부분으로서 생성된 해의 수는 노력의 대가 또는 컴퓨터의 계산 부담량으로 간주하여 그에 상응하는 상대오차를 검토해야 한다. 즉 표1에서 보면 OPTIMAL 알고리듬은 최적해를 구하기 위해  $n!$  번의 계산 노력은 기울인 반면 CAMPBEL, PEIDAN, CAMRING 알고리듬은 휴리스틱 해를 구하기 위해 각각  $(m-1)$ 번, 1번,  $(m-1)$ 번의 계산 노력을 들었다.

#### 4. 휴리스틱 알고리듬의 평가 및 분석

##### 4·1 평가 및 분석 절차

휴리스틱 알고리듬의 유효성을 평가, 분석하는 일반적인 방법은 없으나, 본 연구에서는 최적해와의 상대오차(Relative Error)에 대한 평균값(Mean), 중앙값(Median), 최소값(Minimum), 최대값(Maximum) 등 주요 통계량으로 각 알고리듬을 평가한다. 또 한 가공시간과 준비시간의 비율, 기계의 수, 작업의 수 등 3가지 요인(Factor)에 대한 수준(Level)을 10번 반복하는  $4 \times 3 \times 3$ 으로 설계하여 상대오차에 대한 영향 정도를 알아 보기 위하여 (1) 교호작용이 있는 경우의 분산분석(ANOVA with Interaction), (2) 교호작용이 없는 경우의 분산분석(ANOVA without Interaction), (3)

단순 회귀분석(Simple Regression Analysis), (4) 다중 회귀분석(Multiple Regression Analysis) 등을 SPSS 통계 패키지로 분석하여 보았다.

상대오차는 최적해에 대한 각 휴리스틱 알고리듬 해의 차이를 나타내는 지표로 다음식

$$RE = \frac{T_{HA} - T_{Op}}{T_{Op}} \times 100(\%) \quad \dots \dots \dots \quad (1)$$

에 의해 계산되며 여기서 Top는 최적해의 생산기간을, THA는 휴리스틱 알고리듬에 의한 생산기간을 나타낸다.

상대오차에 대한 영향을 알아 보기 위해 설정된 3가지 요인에 대한 각 수준과 값은 표2와 같다. 여기서 PS는 가공시간과 준비시간의 비율이며 이 시간은

표 2. 각 요인에 대한 수준과 값

요인	수준	값
P S	4	0.5, 1.0, 1.5, 2.0
M	3	4, 8, 10
N	3	5, 6, 7

컴퓨터(IBM PC-AT)를 이용하여 랜덤(Random)하게 일양분포(Uniform Distribution)를 사용하여 생

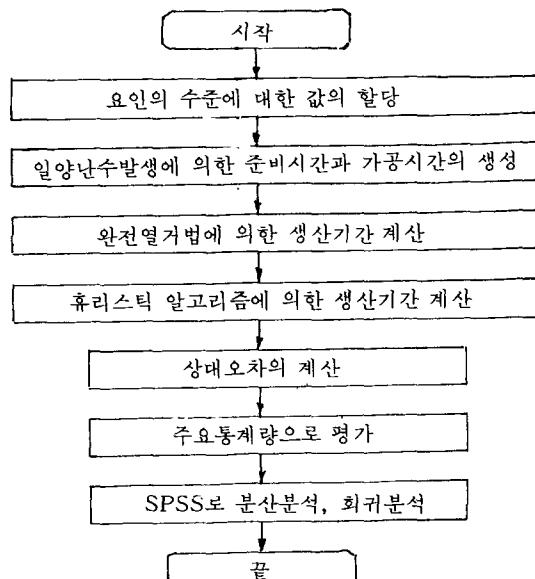


그림 4. 평가 및 분석 절차 흐름도표

표 4. 교호작용이 있는 경우의 분산분석

알고리듬	PS	M	N	PS×M	PS×N	M×N	PS×M×N
CAMPBEL	**	**	-	*	-	-	-
PEIDAN	**	**	**	-	**	*	-
CAMRING	**	**	**	-	-	**	-

단, \*\* : 1% 유의, \* : 5% 유의

성했으며, M는 기계, N는 작업을 나타낸다. 이 실험은 10번 반복하는  $4 \times 3 \times 3$  수준으로 설계되었기 때문에 각 알고리듬에 대한 판찰치는 360개가 된다. 이러한 평가 및 분석 방법에 대한 흐름도표는 그림 4와 같으며 프로그램은 C언어를 사용하여 작성하였다.

#### 4 · 2 평가 및 분석

## ① 주요 통계량

상대오차(RE)는 최적해에 대한 각 휴리스틱 알고리듬의 백분율로 나타낸 것으로 표3과 같다.

표 3. 실험결과에 대한 주요 통계량

(각 샘플수=360, 단위 %)

알고리듬	평균값	중앙값	최소값	최대값
CAMPBEL	4.353	3.470	0.000	40.156
PEIDAN	7.908	7.537	0.000	23.676
CAMRING	8.578	7.182	0.000	29.825

표3을 보면 3가지 알고리듬은 4~8%의 비교적 좋은 평균 상대오차의 값을 가지며 이중에서 CAMPBEL 알고리듬이 가장 유효한 알고리듬으로 평가된다. 특히 PEIDAN 알고리듬은 생성된 해의 수 1번으로 얻은 결과로서  $(m-1)$ 번의 노력으로 얻은 CAMRING 알고리듬 보다도 상대오차가 적게 나와 좋은 알고리듬으로 평가된다.

## ② 부산분석

RE에 대한 PS, M, N의 영향 정도를 평가하기 위해서 교호작용이 있는 경우의 분산분석을 하면 표4와 같다.

표4를 보면 3가지 알고리듬의 상대오차에 공통적으로 영향을 미치는 요인은 PS, M, N이며 CAMP-BELL 알고리듬에서만 N요인이 유의적으로 나오지 않았다. 그러나 CAMPBELL 알고리듬에서의  $PS \times M$ ,  $M \times N$  교호작용과 CAMRING 알고리듬에서의  $M \times N$  교호작용을 제외하고는 대부분의 교호작용이 유의적으로 나오지 않은 것은 상대오차에 영향을 받지 않은 것을 의미한다. 따라서 교호작용을 오차항에 풀링(Pooling)하여 분산분석을 행하면 표5와 같다.

표5를 보면 표4와 달리 CAMPBELL 알고리듬에서 N요인이 유의적으로 나왔다.

### ③ 단순 회귀분석

RE에 대한 PS, M, N의 선형 모델에 대한 상관계 수 및 회귀식을 나타내면 표6, 표7과 같다.

표 5. 교호작용이 없는 경우의 분산분석

알고리듬	PS	M	N
CAMPBEL	**	**	**
PEIDAN	**	**	**
CAMRING	**	**	**

단, \*\* : 1% 유의

표 7. 회귀식

알고리듬	PS	M	N
CAMPBEL	$RE = 19.95645 - 1.35686 \cdot PS$	$RE = 6.09368 - 0.87058 \cdot M$	$RE = 1.65249 + 0.45001 \cdot N$
PEIDAN	$RE = 16.18202 - 0.71949 \cdot PS$	$RE = 10.51740 - 1.30478 \cdot M$	$RE = 2.78095 + 0.85448 \cdot N$
CAMRING	$RE = 32.27103 - 2.06030 \cdot PS$	$RE = 14.17263 - 2.79750 \cdot M$	$RE = 2.71665 + 0.97683 \cdot N$

## (ㄱ) RE와 PS의 관계

3가지 알고리듬 모두 상관계수가 음의 값을 갖는다. 이것은 PS가 작아짐에 따라 RE가 커지는 현상으로 준비시간이 가공시간에 의해 커지면 알고리듬의 성능이 어느 정도 좋지 않은 영향을 받는다는 것을 의미한다. 그러나 PEIDAN 알고리듬에서는 그리 크게 영향을 받지 않는다.

## (ㄴ) RE와 M의 관계

3가지 알고리듬 모두 상관계수가 음의 값을 갖는다. 이것은 M이 커짐에 따라 RE가 작아지는 현상으로 기체의 수가 증가하면 알고리듬의 성능이 향상된다는 것을 의미한다. 특히 CAMRING 알고리듬에서 상관계수의 값이 크게 나온 것은 이 알고리듬이 여러번의 Johnson 알고리듬을 적용하는 기본 구조를 갖고 있어

표 6. 상관계수(유의수준≤5%)

알고리듬	PS	M	N
CAMPBEL	-0.38016	-0.17813	0.09208
PEIDAN	-0.18710	-0.24779	0.16228
CAMRING	-0.40819	-0.40477	0.14134

그만큼 생성하는 해의 수가 많아져 더 좋은 해를 발견해 낼 가능성이 많다는 것을 의미한다.

## (ㄷ) RE와 N의 관계

3가지 알고리듬 모두 양의 상관계수를 갖는다. 이것은 N이 커짐에 따라 RE가 커지는 현상으로 작업의 수가 증가하면 알고리듬의 성능이 그다지 좋지 않은 영향을 받는다는 것을 의미한다.

## ④ 다중 회귀분석

RE에 대한 PS, M, N의 다중 회귀분석에 대한 다중 상관계수와 회귀식은 표8과 같으며 이 변수들간의 RE에 대한 상대적 중요도를 파악하기 위하여 단계적 방법(Stepwise Method)을 적용하여 얻은 상관계수의 설명력은 표9와 같다.

표 8. 다중 상관계수와 회귀식(유의수준&lt;5%)

알고리듬	다중상관계수	회귀식
CAMPBEL	0.42980	$RE = 18.99757 - 1.35686 \cdot PS - 0.87058 \cdot M + 0.45001 \cdot N$
PEIDAN	0.35035	$RE = 42.53489 - 2.27035 \cdot PS - 2.48014 \cdot M + 0.27317 \cdot N$
CAMRING	0.59197	$RE = 32.00504 - 2.06300 \cdot PS - 2.79750 \cdot M + 0.97683 \cdot N$

표 9. 다중 상관계수의 설명력

알고리듬	다중상관계수	PS 상관계수	M 상관계수	N 상관계수
CAMPBEL	0.42980	0.38016	0.03966	-
PEIDAN	0.35035	0.06271	0.24779	0.03985
CAMRING	0.59197	0.40819	0.16586	0.01792

표9를 보면 CAMPBELL 알고리듬과 CAMRING 알고리듬에서는 PS가, PEIDAN 알고리듬에서는 M이, RE에 대한 상대적 중요도가 다른 변수보다도 크게 나타난다.

## 5. 결론

본 논문에서 개발된 휴리스틱 알고리듬의 평가, 분석한 결과를 요약하면 다음과 같다.

(1) CAMPBEL, PEIDAN, CAMRING 알고리듬의 평균 상대오차는 각각 4.353%, 7.908%, 8.578%의 비교적 좋은 결과를 나타내며 이중에서 CAMPBEL 알고리듬이 가장 유효한 알고리듬으로 평가된다.

(2) 분산분석 결과 개발된 3가지 알고리듬의 상대오차는 모두 가공시간과 준비시간의 비율, 기체의 수, 작업의 수 등 3가지 요인에 대해 5% 수준으로 유의적인 값을 갖는다.

(3) 단순 회귀분석 결과 3가지 알고리듬은 준비시

간이 가공시간에 비해 커지면 알고리듬의 성능에 좋지 않은 영향을 미치는 것으로 사료된다.

(4) 단순 회귀분석 결과 3가지 휴리스틱 알고리듬은 기계의 수가 증가하면 알고리듬의 성능이 향상된다. 특히 CAMRING 알고리듬에서 효과가 큰 이유는 생성하는 해의 수가 많아지는 기본 구조를 갖고 있어서 더 좋은 해를 발견해 낼 기회가 많아지기 때문이다.

(5) 단순 회귀분석 결과 3가지 알고리듬은 작업의 수가 커짐에 따라 알고리듬의 성능이 그리 좋지 않은 영향을 받은 것으로 사료된다.

(6) 다중 회귀분석 결과 CAMPBELL 알고리듬과 CAMRING 알고리듬에서는 가공시간과 준비시간의 비율이, PEIDAN 알고리듬에서는 기계의 수가 다른 변수보다도 상대적 중요도가 크게 나타난다.

결론적으로 본 논문에서 개발된 CAMPBELL, PEIDAN, CAMRING의 3가지 휴리스틱 알고리듬들은 4~8%의 평균 상대오차를 가지나 이것은 1번 또는  $(m-1)$ 번의 계산 노력으로 얻은 결과로써 최적해를 구하기 위한  $n!$ 번의 계산 부담에 비해 효율적인 방법이라 할 수 있다. 즉 본 논문에서는 최적해에 비해 경 확도도 그리 떨어지지 않으면서 손쉽게 구할 수 있는 알고리듬을 개발하였다. 또한 CAMRING 알고리듬은 공정이 긴 다단계 흐름작업에서 유용한 알고리듬으로 평가된다.

개발된 휴리스틱 알고리듬의 성능 향상을 위한 방안으로 네이버후드 탐색 기법을 적용하여 평가, 분석하는 것이 앞으로의 과제로 남는다.

#### References

1. Johnson, S. M., "Optimal Two and Three Stage Production Schedules with Set-Up Times Included," Naval Research Logistics Quarterly, 1(1), pp. 61-68, 1954.
2. Heller, J., "Some Numerical Experiments for an  $M \times J$  Flow Shop and its Decision-Theoretical Aspects," Operations Research, 8(2), pp. 178-184, 1960.
3. Page, E. S., "An Approach to the Scheduling of Jobs on Machines," Operations Research, 23(2), pp. 484-492, 1961.
4. Dudek, R. A., and Teuton, O.F., "Development of M-Stage Decision Rule for Scheduling  $n$  Jobs through M Machines," Operations Research, 12(3), pp. 471-497, 1964.
5. 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, 16(1), pp. 101-107, 1965.
6. Smith, R. D., and Dudek, R. A., "A General Algorithm for Solution of the  $n$ -Job,  $M$ -Machine Sequencing Problem of the Flow Shop," Operations Research, 15(1), pp. 71-82, 1967.
7. Petrov, V.A., Flowline Group Production Planning, Business Publication Limited, 1968.
8. Campbell, H. G., Dudek, R. A., and Smith, M. L., "A Heuristic Algorithm for the  $n$  Job,  $m$  Machine Sequencing Problem," Management Science, 16(4), pp. B630-B637, 1970.
9. Gupta, J. N. D., "A Functional Heuristic Algorithm for the Flowshop Scheduling Problem," Operational Research Quarterly, 22(1), pp. 39-47, 1971.
10. Gupta, J. N. D., "Heuristic Algorithms for Multistage Flowshop Scheduling Problem," AIIE Transactions, 4(1), pp. 11-18, 1972.
11. Bonney, M. C., and Gundry, S. W., "Solutions to the Constrained Flowshop Sequencing Problem," Operational Research Quarterly, 27(4), pp. 869-883, 1976.
12. Dannenbring, D. G., "An Evaluation of Flow Shop Sequencing Heuristics," Management Science, 23(11), pp. 1174-1182, 1977.
13. Gelders, L. F., and Sambandam, N., "Four Simple Heuristics for Scheduling a Flow-Shop," International Journal of Production Research, 16(3), pp. 221-231, 1978.
14. King, J. R., and Spachis, A. S., "Heuristics for Flow-Shop Scheduling," International Journal of Production Research, 18(3), pp. 345-357, 1980.
15. Nawaz, M., Enscore, E. E., and Ham, I., "A Heuristic Algorithm for the  $n$ -Machine,  $m$ -Job Flow-Shop Sequencing Problem," Omega, 11(1), pp. 91-95, 1983.
16. Park, Y. B. Pedgen, C. D., and Enscore, E. E., "A Survey and Evaluation of static Flowshop Scheduling Heuristics," International Journal of Production Research, pp. 127-141, 1984.
17. Cho, G. G., Oh, S. C., and Yang, T. Y., "Operations Scheduling for Multi-Item, Small-Sized Production," Journal of the Korean Institute of Industrial Engineering, 11(2), pp. 57-72, 1985.
18. Uskup, E., and Smith, S. B., "A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems," Operations Research, 23(1), pp. 118-136, 1975.
19. Gupta, J. N. D., and Darrow, W. P., "The Two-

- Machine Sequence Dependent Flowshop Scheduling Problem," European Journal of Operational Research, 24, pp. 439-446, 1986.
20. Szwarc, W., and Gupta, J. N. D., "A Flow-Shop Problem with Sequence-Dependent Additive Setup Times," Naval Research Logistics Quarterly, 34, pp. 619-627, 1987.
21. Srikar, B. N., and Soumen, G., "A MILP Model for the n-Job, M-Stage Flowshop with Sequence Dependent Set-Up Times," International Journal of Production Research, 24(6), pp. 1459-1474, 1986.
22. Gupta, S. K., "n jobs m machines Job-Shop Problems with Sequence-Dependent Set-Up Times, International Journal of Production Research, 20(5), pp. 643-656, 1982.
23. Gupta, J. N. D., "Flowshop Schedules with Sequence Dependent Setup Times," Journal of the Operations Research Society of Japan, 29(3), pp. 206-219, 1986.
24. Nie, N. H., Hull, C. H., Jenkins, J. G., Steinbrenner, K., and Bent, D. H., Statistical Package for the Social Sciences, 2nd ed., McGraw-Hill, New York, 1975.