

준비시간이 종속적인 n/M 스케줄링 문제의 휴리스틱 알고리즘(I)

A Development of Heuristic Algorithms for the Multi-stage Manufacturing Systems with Sequence Dependent Setup Times

崔 聖 雲*
魯 仁 珪**

ABSTRACT

This paper is concerned with a development and evaluation of heuristic algorithms for the n -job, M -stage flowshop with sequence dependent setup times.

Three heuristic algorithms, CAIDAN, DANNEN and PETROV, are proposed. The makespan is taken as a performance measure for the algorithms. The experiment for each algorithm is designed for a $4 \times 3 \times 3$ factorial design with 360 observations. The experimental factors are PS (ratio of processing times to setup times), M (number of machines), and N (number of jobs).

The makespan of the proposed heuristic algorithms is compared with the optimal makespan obtained by the complete enumeration method. The result of comparison of performance measure is called a relative error. The mean relative errors of CAIDAN, DANNEN and PETROV algorithms are 4.488%, 6.712% and 7.282%, respectively. The computational results are analysed using SPSS.

The experimental results show that the three factors are statistically significant at 5% level.

* 경원대학교 산업공학과 전임강사
** 한양대학교 산업공학과 교수

1. 서 론

실제로 대부분의 현상에서는 어떤 작업에 대한 준비시간이 바로 전에 끝낸 작업의 종류에 따라 영향을 받는다. 그런 상황하에서는 대부분의 스케줄링 연구에서 가정하고 있듯이 준비시간을 가공시간에 포함시키는 것은 옳지 못하며 준비시간이 작업순서에 종속적인 스케줄링 문제로서 고려되어야 한다.

특히 기계 1대인 경우에 준비시간이 작업순서에 종속적인 스케줄링 문제($n/1/D/F/F_{max}$)는 1명의 순회 판매원 문제(1TSP : Single-Traveling Salesman Problem)와 같은 유형의 문제로서 준비시간의 합을 최소화하는 것과 같다. 이러한 1단계 1기계 SDST(Sequence Dependent Setup Times) 스케줄링 문제에 대한 휴리스틱 알고리즘은 1TSP의 작업순서의 처음과 마지막에 해당하는 작업을 가상 작업(Dummy Job)으로 하여 연구하였다[1-3].

또한 병렬 기계 m 대인 경우에 준비시간이 작업순서에 종속적인 스케줄링 문제($n/Identical-m/D/F/F_{max}$)는 m 명의 순회 판매원 문제(MTSP : Multi-Traveling Salesman Problem)와 같은 유형의 문제로서 역시 준비시간의 합을 최소화하는 것이 수행 기준이 된다. 이러한 1단계 m 기계 SDST 스케줄링 문제에 대한 알고리즘은 MTSP의 기법을 직접 응용하여 연구하였다[4-8].

그러나 본 논문에서는 n 개의 작업이 서로 다른 m 대의 기계에서 똑같은 순서로 가공될 때 각 기계의 준비시간이 작업순서에 종속적인 경우($n/m/D/F/F_{max}$)의 생산기간(Makespan)을 최소화하는 스케줄링 문제를 다루기로 한다. 여기서 총 작업시간이란 가공시간과 준비시간의 합을 의미한다. 이러한 2단계 m 기계 SDST 스케줄링 알고리즘[9-11] 및 M 단계 m 기계 스케줄링 알고리즘[12]은 어느 정도 연구되었으

나 특히 M 단계 m 기계 스케줄링 알고리즘은 계산량이 과중한 최적화 기법을 사용하여 알고리즘을 연구하였다. Gupta[13]는 $n/m/D/J/F_{max}$ 스케줄링 문제 즉 잡·샵(Job·Shop)인 경우의 분기한계(Branch and Bound)기법을 연구하였으며, Gupta[14]는 제한된 재공품 보관장소가 있는 경우의 $n/m/D/F/F_{max}$ 스케줄링 문제에서의 휴리스틱 알고리즘을 연구하였다. 그러나 작업순서가 준비시간에 종속적인 M 단계 스케줄링 문제에 대해서는 보다 효율적인 알고리즘을 연구할 필요가 있다. 따라서 본 논문에서는 $n/m/I/F/F_{max}$ 에서 적용했던 휴리스틱 알고리즘[15-29]을 기초로 하여 $n/m/D/F/F_{max}$ 에서 적용할 수 있는 3가지의 새로운 휴리스틱 알고리즘을 연구·개발하기로 한다.

또한 개발된 3가지의 휴리스틱 알고리즘은 완전열거법으로 구한 최적해와의 상대오차(Relative Error)를 구하여 평가하였으며(4~7%), 가공시간과 준비시간의 비율, 기계의 수, 작업의 수 등의 실험조건에 따른 상대오차의 변화를 SPSS 컴퓨터 패키지[30]로 분석하였다.

2. $n/m/D/F/F_{max}$ 스케줄링 문제

2-1. 문제의 가정

본 논문에서 연구하고자 하는 문제는 준비시간이 가공시간에 종속적인 경우 홀로우·샵 스케줄링에서 생산기간을 최소화 하는 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 정체상태, 준비상태, 가공상태, 가공 마무리상태 중 하나이다.

(5) 어떠한 작업도 완료하기 전에는 시스템을 떠날 수 없다. 어떤 경우에도 작업분할, 타부서 이관, 또는 외부하청이 불가능하다.

(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) 모든 작업의 우선 순위는 고정되어 있다.

(10) 다음 기계로의 작업의 이동시간은 무시해도 좋다.

(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 : 작업의 번호 ($j=1, 2, \dots, n$)

i : j 작업의 전 단계 작업 번호 ($i=1, 2, \dots, n$)

l : 보조문제 번호 ($l=1, 2, \dots, a$)

k : 기계의 번호 ($k=1, 2, \dots, m$)

P : 가공시간

r : 작업순서 위치

S : 준비시간

T_{rj1}^l : 첫번째 가상 기계에서의 가공시간과 준

비시간의 합.

T_{rj2}^l : 두번째 가상 기계에서의 가공시간과 준비시간의 합.

A : T_{rj2}^l 이 T_{rj1}^l 을 초과하는 j 의 집합

B : T_{rj1}^l 이 T_{rj2}^l 을 초과하는 j 의 집합

S_q^* : 최소 생산기간의 작업순서(최선 작업순서)

3. 휴리스틱 알고리즘의 개발

3-1. CAIDAN 알고리즘

CAIDAN (Campbell et al. and Inverse Dannenbring) 알고리즘은 Campbell et al. [18] 알고리즘에 Dannenbring[22] 알고리즘의 경사지수를 역(Inverse)으로 가중하여 개발하였다.

CAIDAN 알고리즘에 대한 절차는 다음과 같다.

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

단계 2. 준비시간의 $n \times m$ 행렬 S 를 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.

단계 3. "2대의 기계" 문제로 변형된 보조 문제의 수 a 는 $(m-1)$ 개까지 고려한다.

단계 4. 첫번째 보조 문제는 $l=1$ 로 놓는다.

단계 5. $r(=1, 2, \dots, n)$ 는 작업순서의 위치를 나타내며 (r)은 배정된 작업의 위치이다. 작업이 배정된 첫번째 위치는 $r=1$ 로 놓는다.

단계 6. 모든 작업에 대해서

$$T_{rj1}^l = \sum_{k=1}^l k(P_{jk} + S_{(r-1)jk}) \text{와}$$

$$T_{rj2}^l = \sum_{k=m+1-l}^m k(P_{jk} + S_{(r-1)jk}) \text{를 계산한}$$

다.

단계 7. $A = \{j | T_{rj1} < T_{rj2}\}$ 로 놓는다. 가장 작은 T_{rj1} 을 갖는 작업 j^* 를 r 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. A 가 공집합이면 다음 단계로 가고 그렇지 않으면 단계 9로 간다.

단계 8. $B = \{j | T_{rj1} > T_{rj2}\}$ 로 놓는다. 가장 큰 T_{rj2} 를 갖는 작업 j^* 를 r 위치에 배정한다.

단계 9. $r < n$ 이면 $r = r + 1$ 로 놓고 단계 6으로 가고 $r = n$ 이면 다음 단계로 간다.

단계 10. $l < a$ 이면 $l = l + 1$ 로 놓고 단계 5로 가고 $l = a$ 이면 다음 단계로 간다.

단계 11. 원래의 $n \times m$ 가공시간 행렬과 $n \times m$ 준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계 12. 가장 최소인 생산기간을 갖는 작업순서 S_q^* 를 선택한다. 이러한 CAIDAN 알고리즘의 흐름도표는 그림 1과 같다.

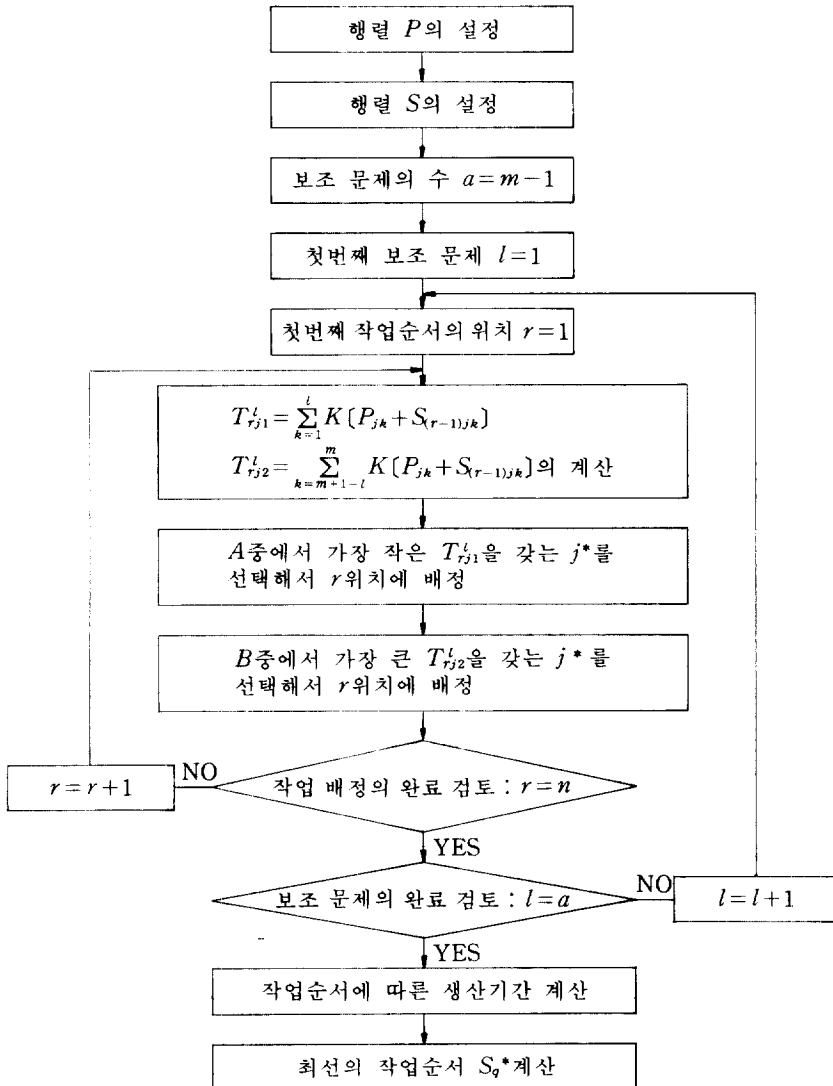


그림 1. CAIDAN 알고리즘의 흐름도표

3-2. DANNEN 알고리즘

DANNEN(Dannenbring) 알고리즘은 빠른 접근 절차(RAP : Rapid Access Procedure)라고도 하며 경사지수와 Campbell et al. 알고리즘과 유사한 가중치를 사용한다.

그러나 DANNEN 알고리즘은 CAIDAN 알고리즘과 달리 보조 문제가 존재하지 않는다.

DANNEN 알고리즘에 대한 절차는 다음과 같다.

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

단계 2. 준비시간의 $n \times m$ 행렬 S 를 설정한다. S_{ijk} 는 k 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.

단계 3. $r(=1, 2, \dots, n)$ 는 작업순서의 위치를 나타내며 (r)은 배정된 작업의 위치이다. 작업이 배정된 첫번째 위치는 $r=1$ 로 놓는다.

단계 4. 모든 작업에 대해서

$$T'_{rj1} = \sum_{k=1}^m (m-k+1) [P_{jk} + S_{r-1jk}] \text{와}$$

$$T'_{rj2} = \sum_{k=1}^m k [P_{jk} + S_{r-1jk}] \text{를 계산한다.}$$

단계 5. $A = \{j | T'_{rj1} < T'_{rj2}\}$ 로 놓는다. 가장 작은 T'_{rj1} 을 갖는 작업 j^* 를 r 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. A 가 공집합이면 다음 단계로가고 그렇지 않으면 단계7로 간다.

단계 6. $B = \{j | T'_{rj1} > T'_{rj2}\}$ 로 놓는다. 가장 큰 T'_{rj2} 를 갖는 작업 j^* 를 r 위치에 배정한다.

단계 7. $r < n$ 이면 $r=r+1$ 로 놓고 단계4로 가고 $r=n$ 이면 다음 단계로 간다.

단계 8. 원래 $n \times m$ 가공시간 행렬과 $n \times m$ 준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계 9. 가장 최소인 생산기간을 갖는 작업순

서 S_q^* 를 선택한다.

이러한 DANNEN 알고리즘의 흐름도표는 그림2와 같다.

3-3. PETROV 알고리즘

PETROV(Petrov[17]) 알고리즘은 2대 기계인 홀로우·샵에서 생산기간을 최소화하는 Johnson 알고리즘의 확장으로 기계가 짝수일 경우와 홀수일 경우를 구분하여 알고리즘을 개발하였다.

PETROV 알고리즘에 대한 절차는 다음과 같다.

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

단계 2. 준비시간의 $n \times m$ 행렬 S 를 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 대하여 $S_{ijk}=0$ 이다.

단계 3. $r(=1, 2, \dots, n)$ 는 작업순서의 위치를 나타내며 (r)은 배정된 작업의 위치이다. 작업이 배정된 첫번째 위치는 $r=1$ 로 놓는다.

단계 4. 모든 작업에 대해서

$$T'_{rj1} = \sum_{k=1}^{k_1} [P_{jk} + S_{r-1jk}] \text{와}$$

$$T'_{rj2} = \sum_{k=k_2}^m [P_{jk} + S_{r-1jk}] \text{를 계산한다.}$$

단, m 이 짝수이면 $k_1 = \lfloor \frac{m}{2} \rfloor$, $k_2 = \lfloor \frac{m}{2} \rfloor + 1$ 이며, m 이 홀수이면 $k_1 = k_2 = \lfloor \frac{m}{2} \rfloor + 1$ 이다.

단계 5. $A = \{j | T'_{rj1} < T'_{rj2}\}$ 로 놓는다. 가장 작은 T'_{rj1} 을 갖는 작업 j^* 를 r 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. A 가 공집합이면 다음 단계로가고 그렇지 않으면 단계7로 간다.

단계 6. $B = \{j | T'_{rj1} > T'_{rj2}\}$ 로 놓는다. 가장 큰 T'_{rj2} 를 갖는 작업 j^* 를 r 위치에 배정한다.

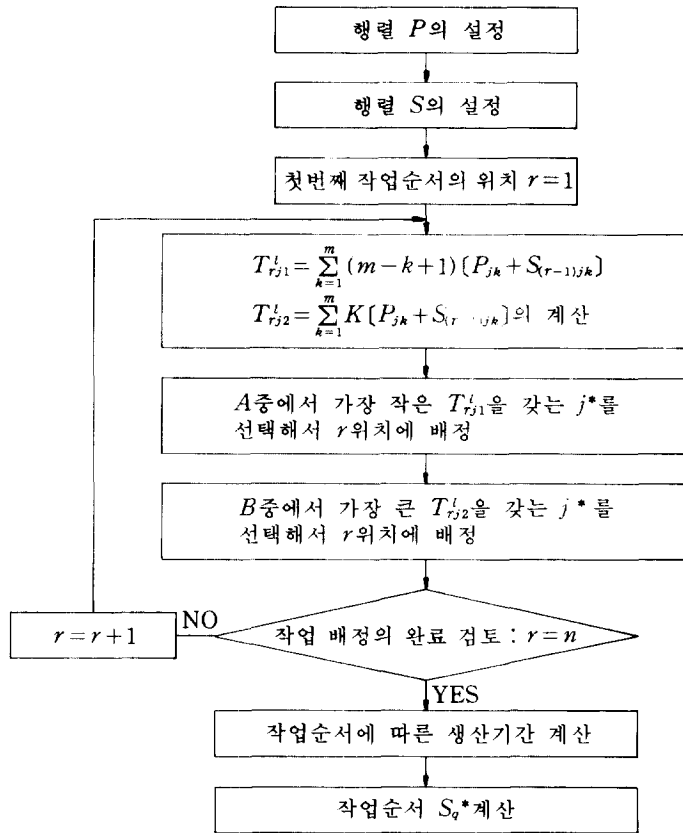


그림 2. DANNEN 알고리즘의 흐름도표

단계 7. $r < n$ 이면 $r = r + 1$ 로 놓고 단계 6으로 가고 $r = n$ 이면 다음 단계로 간다.

단계 8. 원래의 $n \times m$ 가공시간 행렬과 $n \times m$ 준비시간 행렬을 사용하여 배정된 작업순서에 따라 생산기간을 계산한다.

단계 9. 가장 최소인 생산기간을 갖는 작업순서 S_q^* 를 선택한다.

이러한 PETROV 알고리즘의 흐름도표는 그림 3과 같다.

3-4. 알고리즘의 기본 구조

이러한 3알고리즘은 결국 i) Johnson 알고리즘의 적용 횟수 ii) 지수의 사용방법에 따라 기

본 구조가 형성되며 이 변수들은 생성된 해의 수에 영향을 주게 되며 표 1과 같다.

표 1. 각 휴리스틱 알고리즘의 기본 구조

알고리즘	Johnson 알고리즘		지수	생성된 해의 수(개)
	1회	여러번		
CAIDAN	-	×	×	$m - 1$
DANNEN	×	-	×	1
PETROV	×	-	-	1
OPTIMAL	-	-	-	$n!$

*OPTIMAL : 최적 알고리즘

표 1에서 보면 CAIDAN 알고리즘은 Johnson 알고리즘을 $(m - 1)$ 번 적용하여 $(m - 1)$ 개의 해

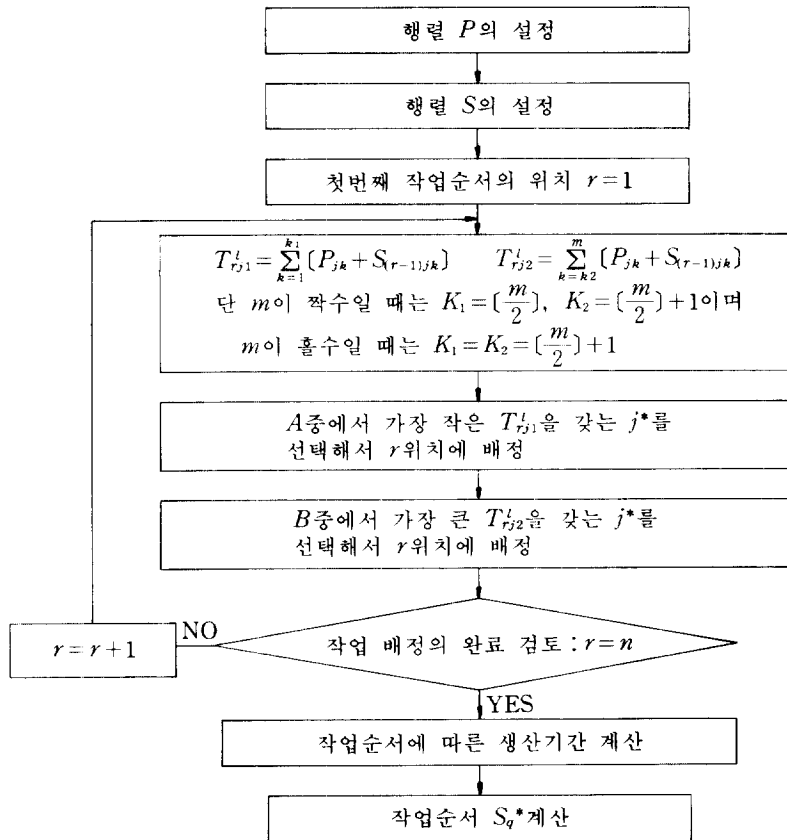


그림 3. PETROV 알고리즘의 흐름도표

를 생성할 때 그 중에서 가장 작은 생산기간을 갖는 작업순서를 결정하며, DANNEN 알고리즘과 PETROV 알고리즘은 Johnson 알고리즘을 1회 적용하여 작업순서를 결정한다. 또한 PETROV 알고리즘에서는 지수를 사용하지 않았으며 CAIDAN 알고리즘과 DANNEN 알고리즘은 지수를 사용하였다. 이 기본 구조는 나중에 최적해와의 상대오차를 평가할 경우 반드시 고려해야 하는 부분으로서 생성된 해의 수는 노력의 댓가 또는 컴퓨터의 계산 부담량으로 간주하여 그에 상응하는 상대오차를 검토해야 한다. 즉 표1에서 보면 OPTIMAL 알고리즘은 최적해를 구하기 위해 $n!$ 번의 계산 노력이 들며 CAIDAN, DANNEN, PETROV 알고리즘은

각각 휴리스틱 해를 구하기 위해 $(m-1)$ 번, 1 번, 1 번씩의 계산 노력이 든다.

4. 휴리스틱 알고리즘의 평가 및 분석

4-1. 평가 및 분석 절차

휴리스틱 알고리즘의 유효성을 평가하기 위해서 본 연구에서는 최적해와의 상대오차(Relative Error)에 대한 평균값(Mean), 중앙값(Median), 최소값(Mimum), 최대값(Maximum) 등 주요 통계량(Statistics)으로 각 알고리즘을 평가한다. 또한 가공시간과 준비시간의

비율, 기계의 수, 작업의 수 등 3가지 요인 (Factor)에 대한 수준(Level Value)을 10번 반복하는 $4 \times 3 \times 3$ 으로 설계하여 상대오차에 대한 영향 정도를 알아 보기 위하여 i) 교호작용이 있는 경우의 분산분석(ANOVA with Interaction), ii) 교호작용이 없는 경우의 분산분석(ANOVA without Interaction), iii) 단순회귀분석(Simple Regression Analysis), iv) 다중 회귀분석(Multiple Regression Analysis) 등을 SPSS 통계 패키지로 분석하여 보았다.

상대오차는 최적해에 대한 각 휴리스틱 알고리즘해의 차이를 나타내는 지표로 다음식

$$RE = \frac{T_{HA} - T_{OP}}{T_{OP}} \times 100(\%)$$

에 의해 계산되며 여기서 T_{OP} 는 최적해의 생산기간을, T_{HA} 는 휴리스틱 알고리즘에 대한 생산기간을 나타낸다.

상대오차에 대한 영향을 알아보기 위해 설정된 3가지 요인에 대한 각 수준과 값은 표2와 같다. 여기서 PS 는 가공시간과 준비시간의 비율이며 이 시간은

표 2. 각 요인에 대한 수준과 값

요인	수준	값
PS	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 생성했으며, M 는 기계, N 는 작업을 나타낸다. 이 실험은 10번 반복하는 $4 \times 3 \times 3$ 수준으로 설계되었기 때문에 각 알고리즘에 대한 관찰치는 360개가 된다. 이러한 평가 및 분석방법에 대한 흐름도표는 그림4와 같으며 프로그램은 C언어를 사용하여 작성하였다.

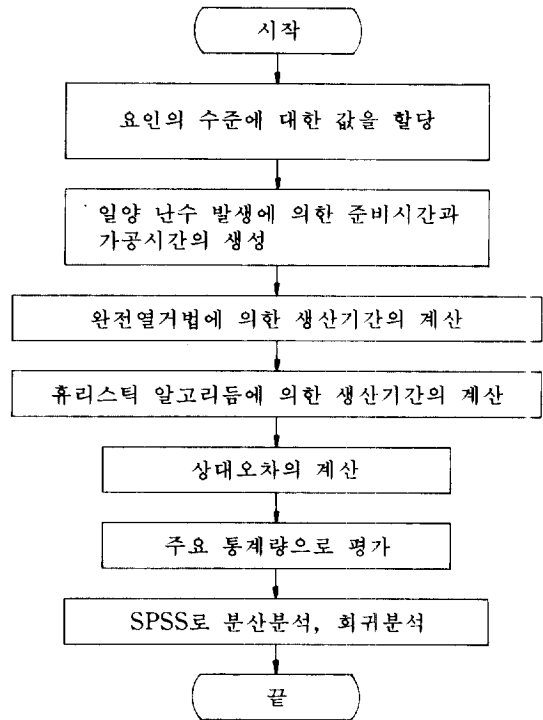


그림 4. 평가 및 분석 절차 흐름도표

4-2. 평가 및 분석 결과

①. 주요 통계량

상대오차(RE)는 최적해에 대한 각 휴리스틱 알고리즘의 해와의 차이를 백분율로 나타낸 것으로 표3과 같다.

표 3. 실험결과에 대한 주요 통계량
(각 샘플수=360)

알고리즘	평균값 (%)	중앙값 (%)	최소값 (%)	최대값 (%)
CAIDAN	4.488	4.226	0.000	18.696
DANNEN	6.712	6.026	0.000	27.136
PETROV	7.282	6.311	0.000	28.261

표 3을 보면 3가지 알고리즘은 4~7%의 비교

적 좋은 평균 상대오차의 값을 가지며 이 중에서 CAIDAN 알고리즘이 가장 유효한 알고리즘으로 평가된다. 특히 DANNEN 알고리즘과 PETROV 알고리즘은 생성된 해의 수 1번으로 얻은 결과로써 계산 노력에 비하여 상대오차가

상당히 좋게 나온 것으로 평가된다.

②. 분산분석

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

표 4. 교호작용이 있는 경우의 분산분석

알고리즘	PS	M	N	PS × M	PS × N	M × N	PS × M × N
CAIDAN	**	**	**	-	-	**	-
DANNEN	**	**	-	-	-	-	-
PETROV	**	**	-	**	-	-	-

단, **: 1% 유의

표4를 보면 3가지 알고리즘의 상대오차에 공통적으로 영향을 미치는 요인은 PS와 M이며, CAIDAN 알고리즘에서는 N요인도 유의적으로 나왔다. 그러나 CAIDAN 알고리즘에서의 M × N 교호작용과 PETROV 알고리즘에서의 PS × M 교호작용을 제외하고는 대부분의 교호작용이 유의적으로 나오지 않은 것은 상대오차에 영향을 주지 않는 것을 의미한다. 따라서 교호작용을 오차항에 풀링(Pooling)하여 분산분석을 행하면 표5와 같다.

③. 단순 회귀분석

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

표 6. 상관계수(유의수준 ≤ 5%)

알고리즘	PS	M	N
CAIDAN	-0.12419	-0.31433	0.30031
DANNEN	-0.42807	-0.13189	-
PETROV	-0.40924	0.15753	-

표 5. 교호작용이 없는 경우의 분산분석

알고리즘	PS	M	N
CAIDAN	**	**	**
DANNEN	**	**	-
PETROV	**	**	-

단, **: 1% 유의

표5를 보면 앞의 표4와 같은 평가를 내릴 수 있다.

i) RE와 PS의 관계

3알고리즘 모두 상관계수가 음의 값을 갖는다. 이것은 PS가 작아짐에 따라 RE가 커지는 현상으로 준비시간이 가공시간에 비해 커지면 알고리즘의 성능이 그다지 좋지 않은 영향을 받는다는 것을 의미한다. 그러나 CAIDAN 알고리즘에서는 그리 크게 영향을 받지 않는다.

ii) RE와 M의 관계

3알고리즘 모두 상관계수가 음의 값을 갖는

표 7. 회귀식

알고리즘	<i>PS</i>	<i>M</i>	<i>N</i>
CAIDAN	$RE = 8.27346 - 0.32917 \cdot PS$	$RE = 6.76962 - 1.14080 \cdot M$	$RE = -2.05141 + 1.08990 \cdot N$
DANNEN	$RE = 26.62382 - 1.73143 \cdot PS$	$RE = 8.17336 - 0.73049 \cdot M$	-
PETROV	$RE = 26.58678 - 1.67866 \cdot PS$	$RE = 9.05173 - 0.88477 \cdot M$	-

다. 이것은 *M*이 커짐에 따라 *RE*가 작아지는 현상으로 기계의 수가 증가하면 알고리즘의 성능이 향상된다는 것을 의미한다. 특히 CAIDAN 알고리즘에서 상관계수 값이 크게 나온 것은 이 알고리즘이 생성하는 해의 수가 많아지는 기본 구조를 갖고 있기 때문에 더 좋은 해를 발견해 낼 기회가 많다는 것을 의미한다.

iii) *RE*와 *N*의 관계

3알고리즘 중 CAIDAN에서만 양의 상관계수를 갖는다. 이것은 *N*이 커짐에 따라 *RE*가 커

지는 현상으로 작업의 수가 증가하면 알고리즘의 성능이 어느정도 좋지 않은 영향을 받는다는 것을 의미한다.

④. 다중 회귀분석

*RE*에 대한 *PS*, *M*, *N*의 다중 회귀분석에 대한 다중 상관계수와 회귀식은 표8과 같으며 이 변수들 간의 *RE*에 대한 상대적 중요도를 파악하기 위하여 단계적 방법(Stepwise Method)을 적용하여 얻은 다중 상관계수의 설명력은 표9와 같다.

표 8. 다중 상관계수와 회귀식(유의수준 ≤ 5%)

알고리즘	다중 상관계수	회 귀 식
CAIDAN	0.45212	$RE = 4.01565 - 0.32917 \cdot PS - 1.14080 \cdot M + 1.08990 \cdot N$
DANNEN	0.44924	$RE = 26.94273 - 1.73143 \cdot PS - 0.73049 \cdot M + 1.19035 \cdot N$
PETROV	0.43934	$RE = 10.49618 - 0.82987 \cdot PS - 0.51784 \cdot M + 0.74528 \cdot N$

표 9. 다중 상관계수의 설명력

알고리즘	다중 상관계수	<i>PS</i> 상관계수	<i>M</i> 상관계수	<i>N</i> 상관계수
CAIDAN	0.45212	0.01739	0.31433	0.12040
DANNEN	0.44924	0.42807	0.01985	-
PETROV	0.43934	0.40924	0.02927	-

표9를 보면 CAIDAN 알고리즘에서 *M*이, DANNEN 알고리즘과 PETROV 알고리즘에서

는 *PS*가, *RE*에 대한 상대적 중요도가 다른 변수 보다도 크게 나타난다.

5. 결 론

본 논문에서 개발된 3가지 알고리즘을 평가·분석한 결과를 요약하면 다음과 같다.

i) 본 논문에서 개발한 CAIDAN, DANNEN, PETROV 알고리즘의 평균 상대오차는 각각 4.488%, 6.712%, 7.282%로서 좋은 결과를 나타내었다.

ii) 분산분석 결과 개발된 3가지 휴리스틱 알고리즘의 상대오차는 가공시간과 준비시간의 비율, 기계의 수, 작업의 수 등 3가지 요인에 대해 5% 수준에서 유의적인 값을 갖는다.

iii) 단순 회귀분석 결과 3가지 휴리스틱 알고리즘은 준비시간이 가공시간에 비해 커지면 알고리즘의 성능이 좋지 않은 영향을 얻는 것으로 사료된다.

iv) 단순 회귀분석 결과 개발된 휴리스틱 알고리즘은 기계의 수가 증가하면 알고리즘의 성능이 향상되며 특히 CAIDAN 알고리즘에서는 알고리즘의 기본 구조상 더 좋은 해를 발견해 낼 수 있는 기회가 많기 때문에 더욱 더 큰 효과

를 보여준다.

v) 단순 회귀분석 결과 CAIDAN 알고리즘은 작업의 수가 증가하면 알고리즘의 성능이 좋지 않은 영향을 받는 것으로 사료된다.

vi) 다중 회귀분석 결과 CAIDAN 알고리즘에서는 기계가, DANNEN·PETROV 알고리즘에서는 가공시간과 준비시간의 비율이 다른 변수보다도 상대적 중요도가 높은 것으로 나타난다.

결론적으로 개발된 3가지 휴리스틱 알고리즘의 평균 상대오차는 4~7% 정도로 나타났지만 이 결과는 1번 또는 $(m-1)$ 번의 계산 노력으로 얻은 결과로써 기존 최적 알고리즘의 $n/$ 번의 계산 부담에 비해 실제적인 면에서 사용 가치가 높다고 할 수 있다. 특히 CAIDAN 알고리즘은 공정이 긴 다단계 흐름공정(Multi-stage Flow Process)에서 보다 유용하다고 할 수 있다.

본 논문에서 제시한 휴리스틱 알고리즘을 기초로 계산 부담을 줄이면서도 상대오차 값이 좋게 나오는 알고리즘을 개발하는 것이 앞으로의 과제로 남는다.

參 考 文 獻

1. Karg, R.L., and Thompson, G.L. (1964), "A Heuristic Approach to Solving Traveling Salesman Problems," Management Science, Vol.10, No.2, pp.225~248.
2. Gavett, J.W. (1965), "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility," Management Science, Vol.11, No.8, pp.B166~176.
3. White, C.H., and Wilson, R.C. (1977), "Sequence Dependent Set-up Times and Job Sequencing," International Journal of Production Research, Vol.15, No.2, pp.191~202.
4. Svestka, J.A., and Huckfeldt, V.E. (1973), "Computational Experience with an M-Salesmen Traveling Salesman Algorithm," Management Science, Vol.

- 19, No. 7, pp. 790~799.
5. Corwin, B. D., and Esogbue, A. O. (1974), "Two Machine Flow Shop Scheduling Problems with Sequence Dependent Set-Up Times: A Dynamic Programming Approach," Naval Research Logistics Quarterly, Vol. 21, pp. 515~524.
 6. Deane, R. H., and White, E. R. (1975), "Balancing Workloads and Minimizing Set-Up Costs in the Parallel Processing Shop," Operational Research Quarterly, Vol. 26, No. 1, pp. 45~53.
 7. Bulfin, R. L., and Parker, R. G. (1976), "On a Two Facility Scheduling Problem with Sequence Dependent Processing Time," AIIE Transactions, Vol. 8, No. 2, pp. 202~209.
 8. Parker, R. G., Deane, R. H., and Holmes, R. A. (1977), "On the Use of a Vehicle Routing Algorithm for the Parallel Processor Problem with Sequence Dependent Changeover Costs," AIIE Transactions, Vol. 9, No. 2, pp. 155~160.
 9. Uskup, E., and Smith, S. B. (1975), "A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems," Operations Research, Vol. 23, No. 1, pp. 118~136.
 10. Gupta, J. N. D., and Darrow, W. P. (1986), "The Two-Machine Sequence Dependent Flow Shop Scheduling Problem," European Journal of Operational Research, Vol. 24, pp. 439~446.
 11. Szwarcz, W., and Gupta, J. N. D. (1987), "A Flow-Shop Problem with Sequence-Dependent Additive Setup Times," Naval Research Logistics Quarterly, Vol. 34, pp. 619~627.
 12. Srikar, B. N., and Soumen, G. (1986), "A MILP Model for the n -Job, M -Stage Flowshop with Sequence Dependent Set-Up Times," International Journal of Production Research, Vol. 24, No. 6, pp. 1459~1474.
 13. Gupta, S. K. (1982), " n Jobs, m Machines Job-Shop Problems with Sequence-Dependent Set-Up Times," International Journal of Production Research, Vol. 20, No. 5, pp. 643~656.
 14. Gupta, J. N. D. (1986), "Flowshop Schedules with Sequence Dependent Setup Times," Journal of the Operations Research Society of Japan, Vol. 29, No. 3, pp. 206~219.
 15. Johnson, S. M. (1954), "Optimal Two and Three Stage Production Schedules with Set-Up Times Included," Naval Research Logistics Quarterly, Vol. 1, No. 1, pp. 61~68.
 16. Palmer, D. S. (1965), "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, pp. 101~107.
 17. Petrov, V. A. (1968), Flowline Group Production Planning, Business Publication Limited.
 18. Campbell, H. G., Dudek, R. A., and Smith, M. L. (1970), "A Heuristic

- Algorithm for the n Job, m Machine Sequencing Problem,*" Management Science, Vol.16, No.4, pp.B630~B637.
19. Gupta, J.N.D. (1971), "*A Functional Heuristic Algorithm for the Flowshop Scheduling Problem,*" Operational Research Quarterly, Vol.22, No.1, pp. 39~47.
 20. Gupta, J. N. D. (1972), "*Heuristic Algorithms for Multistage Flowshop Scheduling Problem,*" AIIE Transactions, Vol.4, No.1, pp.11~18.
 21. Bonney, M.C., and Gundry, S.W. (1976), "*Solutions to the Constrained Flowshop Sequencing Problem,*" Operational Research Quarterly, Vol.27, No.4, pp.869~883.
 22. Dannenbring, D.G. (1977), "*An Evaluation of Flow Shop Sequencing Heuristics,*" Management Science, Vol.23, No.11, pp.1174~1182.
 23. Gelders, L.F., and Sambandam, N. (1978), "*Four Simple Heuristics for Scheduling a Flow-Shop,*" International Journal of Production Research, Vol.16, No.3, pp.221~231.
 24. King, J.R., and Spachis, A.S. (1980), "*Heuristics for Flow-Shop Scheduling,*" International Journal of production Research, Vol.18, No.3, pp.345~357.
 25. Nawaz, M., Enscore, E.E., and Ham, I. (1983), "*A Heuristic Algorithm for the n-Machine, m-Job Flow-Shop Sequencing Problem,*" Omega, Vol.11, No.1, pp.91~95.
 26. Park, Y.B., Pedgen, C.D., and Enscore, E.E. (1984), "*A Survey and Evaluation of Static Flowshop Scheduling Heuristics,*" International Journal of Production Research, Vol.22, No.1, pp.127~141.
 27. Cho, G.G., Oh, S.C., and Yang, T.Y. (1985), "*Operations Scheduling for Multi-Item, Small-Sized Production,*" Journal of the Korean Institute of Industrial Engineers, Vol.11, No.2, pp. 57~72.
 28. Turner, S., and Booth, D. (1987), "*Comparisian of Heuristics for Flow Shop Sequencing,*" Omega, Vol.15, No. 1, pp.75~85.
 29. Hundal, T.S., and Rajgopal, J. (1988), "*An Extension of Palmer's Heuristic for the Flow Shop Scheduling Problem,*" International Journal of Production Research, Vol.26, No.6, pp. 1119~1124.
 30. Nie, N.H., Hull, C.H., Jenkins, J. G., Steinbrenner, K., and Bent, D.H. (1975), *Statistical Package for the Social Sciences, Second Edition,* McGraw-Hill, New York.