Flow Shop에서 준비시간이 작업순서에 종속적인 경우의 휴리스틱 알고리듬의 개발

A Heuristic Algorithm for $n/m/D/F/F_{max}$

최 성 운*

ABSTRACT

This paper is to develop four heuristic algorithms for $n/m/D/F/F_{max}$. A study present numerical example for (H1) algorithm. Among the sequence generated, the best sequence is J2, J1, J6, J7, J4, J3, J5 and makespan is 528.

The optimal makespan of this numerical example can be found as 528 as also, and the worst value of heuristic solution is only 7.2% away from it.

1. 서 론

실제 현장에서는 작업의 준비 시간이 바로 전에 끝낸 작업의 형태에 따라 달라진다. 그런 상황에서는 대부분의 스케줄링 연구에서 가정하고 있듯이 준비 시간을 가공 시간에 포함시키는 것은 옳지 못하다. 준비 시간이 작업 순서에 종속적인 스케줄링 문제는 공구나 작업대의 재배열 같은 준비시간을 요하는 조립라인에서 발생한다. 기계 1대인 경우에, 준비시간이 작업 순서에 종속적인 스케줄링 문제($n/1/D//F_{max}$)는 순회 판매원 문제(Traveling salesman problem)의 특별한 경우 Makespan을 최소화 하는 것으로 준비시간의 합을 최소화하는 것과 같다 (Baker[1]). 또한 병렬 기계 m대인 경우에 준비 시간이 작업 순서에 종속적인 스케줄링 문제 ($n/\text{identical-}m/D/F/F_{max}$) 역시 준비 시간의 합을 최소화 하는 것이 수행 기준이 된다.

그러나 본 논문에서는 n개의 작업이 서로 다른 m대의 기계를 똑같은 순서로 가공하거나, 각 기계의 준비 시간이 작업 순서에 종속적인 경우($n/m/D/F/F_{max}$)의 스케줄링 문제를 다루기로 한다.

1965년에 Palmer[4]는 m개 공정의 작업에 대한 경사지수(Slope index)를 정의하여 계산된 지수값의 내림 차순으로 작업 순서를 정하였으며, 1966년 Petrov[5]는 2대의 기계 문제로, Makespan을 최소화는 Johnson 알고 리듬을 이용하여 작업 순서를 정하였다. 1970년 Campbell et al. [2]은 기계가 m대일 때 (m−1)개까지 2대 기계 문제로 생각하여 각각에 대해 Johnson 알고리듬을 적용하여 문제를 해결하였으며, 1977년 Dannenbring[3]은 경사지수와 Campbell et al. 등과 유사한 가중치를 가공 시간에 고려하여 Johnson 알고리듬으로 문제를 해결하였다.

따라서 본 논문에서는 $n/m/I/F/F_{max}$에서의 기준의 휴리스틱 알고리듬을 $n/m/D/F/F_{max}$

* 한양대학교 대학원 (박사과정)

에서도 적용할 수 있도록 변형하여 4가지의 새로운 휴리스틱 알고리듬의 개발을 시도하고자 한다. 그러나 새로이 개발된 휴리스틱 알고리듬의 상호 비교보다는 각 알고리듬의 응용 개발에 촛점을 두었으며, 그 중 개발된 새로운 휴리스틱 알고리듬을 선택하여 간단한 수치예제를 통하여 최적해와 비교하기로 한다.

2. 휴리스틱 알고리듬

(1) 응용된 Campbell et al. 알고리듬 (H1)

단계 1. 가공 시간의 nxm행렬 P를 설정한다. P_{jk}는 $K^{th}(=1,2,\cdots\cdots,m)$ 기계에 대한 $j^{th}(=1,2,\cdots,n)$인 작업의 가공시간이다.

단계 2. 준비시간의 nxm행렬 S를 설정한다. S_{ijk}는 K^{th}기계에서 작업 $i(=1,2,\cdots,n)$가 작업 j^{th}로 변할 경우의 준비 시간이다.

단, $i=j$ 일 때면 모든 K에 대해서 $S_{ijk}=0$이다.

단계 3. "2대의 기계" 문제로 변형된 보조문제의 수 b는 m−1개까지 고려한다.

단계 4. 첫 번째 보조문제는 $h=1$로 놓는다.

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

단계 6. 모든 작업에 대해서

$$Q^h_{rj1}=\Sigma^h_{k=1}[P_{jk}+S(r-1)jk]$$와

$$Q^h_{rj2}=\Sigma^m_{k=m+1-h}[P_{jk}+S(r-1)jk]$$를 계산한다.

단계 7. $U=\{j\mid Q^h_{rj1}<Q^h_{rj2}\}$ 로 놓는다.

가장 작은 Q^h_{rj1} 를 갖는 작업 j^*를 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다. U가 공집합이면 다음 단계로 가고, 그렇지 않으면 단계 9로 간다.

단계 8. $V=\{j\mid Q^h_{rj1}>Q^h_{rj2}\}$ 로 놓는다.

가장 큰 Q^h_{rj2} 를 갖는 작업 j^*를 r위치에 배정한다.

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

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

단계 11. 원래의 nxm 가공 시간 행렬과 nxm준비시간 행렬을 사용하여 배정된 작업순서에 따라 Makespan을 계산한다.

단계 12. 가장 최소인 Makespan을 갖는 작업순서 S^*_q를 선택한다.

(2) 응용된 Campbell et al.과 Dannenbring 혼합 알고리듬 (H2)

이 알고리듬은 Campbell et al. 알고리듬에 Dannenbring의 경사지수를 가미해서 고려한 알고리듬이다. 이 알고리듬의 적용 순서는 (H2)알고리듬과 똑같으나, 단계 6의 Q^h_{rj1} 과 Q^h_{rj2} 의 계산 방법에서 차이가 있다. 그래서 여기서는 단계 6만 기술하기로 한다.

단계 6. 모든 작업에 대해서

$$Q^h_{rj1}=\Sigma^h_{k=1}(m-k+1)[P_{jk}+S(r-1)jk]$$

$$Q^h_{rj2}=\Sigma^m_{k=m+1-h}(m-k+1)[P_{j,m-k+1},\ S_{(r-1)j,m-k+1}]$$를 계산한다.

(3) 응용된 Dannenbring 알고리듬 (H3)

단계 1. 가공시간의 nxm 행렬 p를 설정한다. P_{jk}는 $K^{th}(=1,2,\cdots,m)$ 기계에 대한 $j^{th}(=1,2,\cdots,n)$ 작업의 가공시간이다.

단계 2. 준비시간의 nxm 행렬 S를 설정한다. S_{ijk}는 K^{th}기계에서 작업 $i(=1,2,\cdots,n)$가 작업 j^{th}로 변할 경우의 준비시간이다.

단, $i=j$면 모든 K에 대해서 $S_{ijk}=0$이다.

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

단계 4. 모든 작업에 대해서

$$Q_{rj1}=\Sigma^m_{k=1}(m-k+1)[P_{jk}+S_{(r-1)jk}]$$와

$Q_{rj2}=\Sigma_{k=1}^{m}(k)\,[P_{jk}+S_{(r-1)jk}]$ 를 계산한다.

단계 5. $U=\{j \mid Q_{rj1}<Q_{rj2}\}$ 로 놓는다.

가장 작은 Q_{rj1} 를 갖는 작업 j^* 를 r 위치에 배정한다. 값이 똑같은 경우에는 임의로 하나를 선택한다.

단계 6. $V=\{j \mid Q_{rj1}\geq Q_{rj2}\}$ 로 놓는다.

가장 큰 Q_{rj2} 를 갖는 작업 j^* 를 r 위치에 배정한다.

단계 7. r < n 이면 r = r + 1 로 놓고 단계 4 로 가고, r = n 이면 다음 단계로 간다.

단계 8. 원래의 nxm 가공시간 행렬과 nxm 준비시간 행렬을 사용하여 배정된 작업 순서에 따라 Makespan 을 계산한다.

(4) 응용된 Petrov 와 Palmer 의 혼합 알고리듬 (H 4)

이 알고리듬은 Petrov 알고리듬에 Palmer 의 경사지수를 가미하여 고려한 알고리듬이다. 이 알고리듬의 적용순서는 (H3) 알고리듬과 똑같으나, 단계 4 의 Q_{rj1} 과 Q_{rj2} 의 계산 방법에서 차이가 있다. 그래서, 여기서는 단계 6 만 기술하기로 한다.

단계 6. 모든 작업에 대해서

$$Q_{rj1}=\Sigma_{k=1}^{k_1}\left(\frac{2k-k_1-1}{2}\right)[P_{jk}+S_{(r-1)jk}]$$

$$Q_{rj2}=\Sigma_{k=k_2}^{m}\left(\frac{2k-k_1+1}{2}\right)[P_{jk}+S_{(r-1)jk}],\ m$$

이 홀수일 때

$$=\Sigma_{k=k_2}^{m}\left(\frac{2k-3k_1+1}{2}\right)[P_{jk}+S_{(r-1)jk}],\ m$$

이 짝수일 때

를 계산한다. 단, m이 홀수일 때는 $k_1=k_2=[\frac{m}{2}]+1$ 이고, m이 짝수일 때는 $k_1=[\frac{m}{2}]$, $k_2=[\frac{m}{2}]+1$ 이다.

3. 수 치 예

2 장에서 $n/m/D/F/F_{max}$ 에서 작용할 수 있는 4 가지의 새로운 휴리스틱 알고리듬을 개발하였다. 본 장에서는 그 중 (H1) 알고리듬에 대한 간단한 예제를 통해서, 최적작업순서와 Makespan을 구하여 최적해와 비교한 결과를 검토하기로 한다.

모든 작업은 4 대의 기계를 똑같은 순서로 가공되며, 가공시간과 각 기계에서의 작업준비시간이 표 1,2,3,4,5에 나타나 있다.

표 1. 가 공 시 간

작업 \ 기계	M 1	M 2	M 3	M 4
J 1	25	35	20	40
J 2	15	15	25	35
J 3	50	20	50	20
J 4	70	60	60	50
J 5	30	35	35	15
J 6	15	30	35	30
J 7	85	80	80	105

표 2. 기계 1 에 대한 준비시간

	J 1	J 2	J 3	J 4	J 5	J 6	J 7
J 1	0	2	2	5	6	11	13
J 2	2	0	2	6	7	11	12
J 3	2	2	0	7	7	11	13
J 4	4	6	5	0	3	4	5
J 5	6	6	6	2	0	4	5
J 6	11	12	13	8	9	0	4
J 7	11	12	15	8	7	3	0

:

표 3. 기계 2 에 대한 준비시간

	J 1	J 2	J 3	J 4	J 5	J 6	J 7
J 1	0	3	2	6	7	13	14
J 2	3	0	2	8	9	15	14
J 3	3	3	0	4	5	15	11
J 4	4	5	6	0	4	6	6
J 5	7	7	8	3	0	6	6
J 6	9	10	10	2	3	0	2
J 7	3	5	9	4	5	2	0

표 4. 기계 3에 대한 준비시간

	J1	J2	J3	J4	J5	J6	J7
J1	0	3	3	6	6	5	6
J2	3	0	5	8	7	7	8
J3	4	5	0	8	7	2	3
J4	11	13	15	0	6	8	10
J5	15	15	20	6	0	11	13
J6	2	4	5	16	11	0	2
J7	3	6	7	7	4	2	0

표 5. 기계 4에 대한 준비시간

	J1	J2	J3	J4	J5	J6	J7
J1	0	2	2	6	6	5	6
J2	2	0	2	6	7	7	8
J3	2	2	0	4	5	2	3
J4	4	5	6	0	3	4	5
J5	6	6	6	2	0	4	5
J6	2	4	5	2	3	0	2
J7	3	5	9	4	5	2	0

이 예제를 (H1)알고리듬의 12단계에 의하여 풀면, h = 1인 경우의 작업순서는 J2, J6, J1, J7, J4, J3, J5 이고, Makespan = 566, h = 2인 경우의 작업 순서는 J2, J6, J7, J4, J1, J3, J5에 Makespan = 560이며, h = 3인 경우의 작업 순서는 J2, J1, J6, J7, J4, J3, J5이고 Makespan = 528이다.

결국 가장 좋은 작업 순서는 h = 3인 경우로서 Makespan = 528이며 이것은 최적해에 의한 작업 순서 및 Makespan과 일치한다. 이것을 칸트차트로 표시하면 그림 1과 같다. 결국 (H1) 알고리듬의 최적해에 대한 최악의 이탈정도는

$$\frac{566-528}{528} \times 100 = 7.2\,\%$$이다.

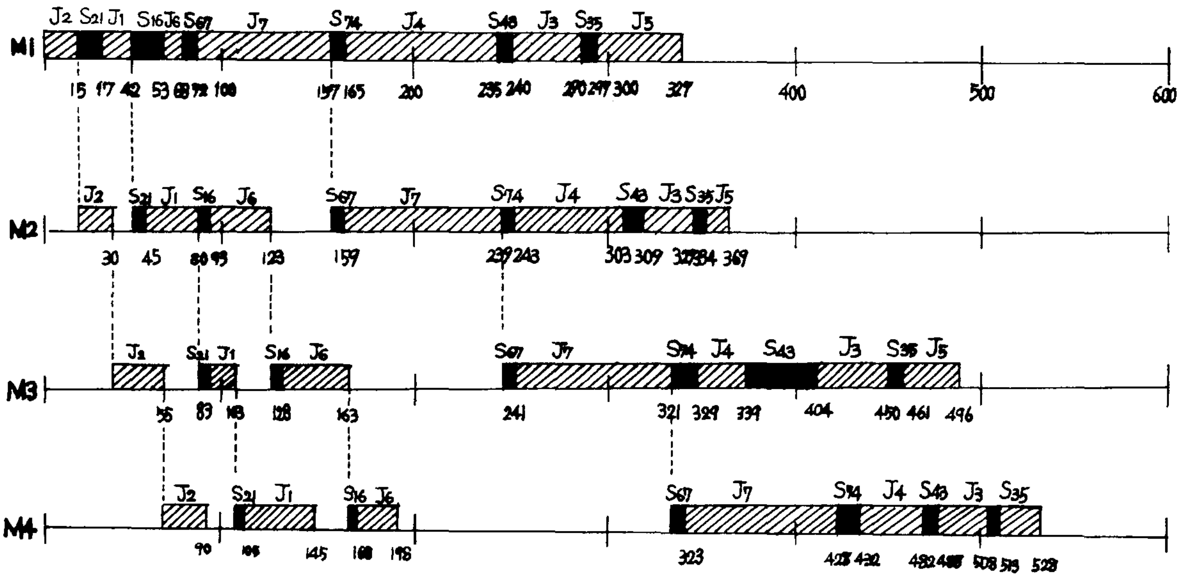

그림 1. 칸 트 차 트

이상으로 상당한 계산의 노력을 줄이면서 최악의 경우에도 최적해와 근사한 해를 얻을 수 있으며, 본 예제에서는 최적해와 똑같은 좋은 결과가 나왔다.

4. 결 론

본 논문에서는 n개의 작업이 서로 다른 m대의 기계를 똑같은 순서로 가공하나, 각 기계에서의 준

비시간이 작업 순서에 종속적인 스케줄링 문제에 대한 새로운 4가지 휴리스틱 알고리듬을 개발하였다.

개발된 알고리듬의 성능을 검토하기 위해 (H1) 알고리듬에 대한 수치 예를 들며 검토한 결과, 최적해와 일치하는 작업 순서 J2, J1, J6, J7, J4, J3, J5와 Makespan = 528을 얻을 수 있었으며, 최악의 경우에도 최적해에 비해 7.2% 정도의 차이만 난다는 것을 알 수 있었다.

또한, 개발된 4가지 휴리스틱 알고리듬을 감도 분석을 통해 최적해와 상호 비교하여 보면서 검토해 보는 것이 앞으로의 과제로 남는다.

참 고 문 헌

(1) Baker, K. R., 1974, Introduction to Sequencing and Scheduling (New York : Wiley).

(2) Campbell, H. G., Dudek, R. A., and Smith, M. L., 1970, An Heuristic Algorithm for the n Job m Machine Sequencing Problem, Management Science, 16, 10.

(3) Dannenbring, D. G., 1977, An Evaluation of Flow-Shop Sequencing Heuristics, Management Science, 23, 11.

(4) Palmer, D. S., 1965, Seoиencing Jobs through a Multi-Stage Process in the Minimum Total Time-A Quick Method of Obtaining a Near Optimum, Operational Research Quarterly, 16, 1.

(5) Petrov, V. A., 1966, Flowline Group Production Planning, (London : Business Publication Ltd.)