

N/M/D/F/Fmax 일정계획 문제에서 최적알고리즘의 개발 -A Development of Optimal Algorithms for N/M/D/F/Fmax Scheduling Problems-

최 성 운*

ABSTRACT

This paper is concerned with the development of optimal algorithms for multi-stage flowshop scheduling problems with sequence dependent setup times.

In the previous researches the setup time of a job is assumed to be able to begin at the earliest opportunity given a particular sequence at the start of operations.

In this paper the setup time of a job is considered to be able to begin only at the completion of that job on the previous machine to reflect the effects of the setup time to the performance measure of sequence dependent setup time flowshop scheduling.

The results of the study consist of two areas; first, a general integer programming(IP) model is formulated and a mixed integer linear programming(MILP) model is also formulated by introducing a new binary variable. Second a depth-first branch and bound algorithm is developed. To reduce the computational burdens we use the best heuristic schedule developed by Choi(1989) as the first trial. The experiments for developed algorithm are designed for a $4 \times 3 \times 3$ factorial design with 360 observations. The experimental factors are PS(ratio of processing time to setup time), M(number of machines), N(number of jobs).

1. 序 論

다단계 제조공정에서 기계의 배치가 부품가공의 순서와 완전히 일치하는 경우 즉, 모든 부품이 동일 기계 가공 순서를 거쳐야 하는 일정계획을 다단계 흐름작업의 일정계획(multi-stage flowshop scheduling)이라 한다.

작업 총 완성시간(makespan)을 최소로 하는 기존의 흐름작업 일정계획에 관한 연구에서는 문제의 복잡성을 줄이기 위해 준비시간의 작업순서에 독립적(SIST: Sequence Independent Setup Time)인 것으로 가정하여 가공시간에 준비시간을 포함시켰다(Ignall et al 1965, Lomnicki 1965, stafford 1988).

그러나 실제 대부분의 공장에서는 한작업에 대한 준비시간이 바로 전에 끝낸 작업의 종류에 따라 영향을 받는다. 이와 같은 경우는 기존 연구에서 가정하고 있는 바와 같이 준비시간을 가공시간에 포함시키는 것은 바람직하지 못하며 준비시간이 작업순서에 종속적(SDST: Sequence Dependent Setup Time)인 것으로 취급되어야 한다. 이러한 상황은 대부분의 가공공정에서 발생하므로 연구의 필요성이 증대되고 있다.

Uskup et. al(1975)은 각 작업에 대한 납기가 존재하는 2단계 공정에서 총 준비시간을 최소화하는 분기한계 알고리즘을 개발하였으며, Gupta(1982)는 주문작업 일정계획(N/M/D/J/Fmax)을 대상으로 하여 아주 작은 문제에서만 적용할 수 있는 분기한계 알고리즘을 제시하였다. Slikar et. al(1986)은 작업이 도착하기 전에 준비를 끝낼 수 있는 것으로 가정하여 혼합정수선형계획으로 정식화하였다.

Gupta et. al(1986)은 2단계 공정에서 효율적으로 사용할 수 있는 4가지의 휴리스틱 알고리즘을 제시하여 평가하였다. 또한, Gupta (1986)는 연속가공을 하거나 제한 혹은 무제한 재공품 보관장소가 있는 경우에 이것을 TSP로 정식화하여 휴리스틱 알고리즘을 개발하였다. Szwarc et. al(1987)은 준비시간이 가법성을 가질 때의

*경원대학교 산업공학과

접수 1990년 4월 25일

휴리스틱 알고리즘을 제시하였다.

이상의 다단계 SDST 흐름작업의 일정계획에 대한 기존문헌을 고찰하면 작업도착 전 준비완료, 연속가공, 무제한 재고품 보관장소 및 준비시간의 가법성등을 가정하여 TSP나 해결하기 쉬운 SISD 흐름작업의 일정계획으로 변형하였다.

그러나 다단계 SDST 흐름작업의 일정계획에서는 준비시간이 작업순서에 따라 변할 수 있으므로 전 단계에서 현 단계에 도착된 작업의 준비시간이 작업 총완성시간에 영향을 크게 미치게 된다. 따라서 본 논문에서는 기존 연구와는 달리 현 단계에서의 준비는 전 단계에서 작업이 도착된 후에 가능한 것으로 가정하여 다음 2가지 분야에 대한 연구를 수행하였다.

첫째, 본 연구에서 제시한 다단계 SDST 흐름작업의 일정계획 모형에 대해 정수계획으로 정식화하였으며 그리고 새로운 이진변수를 도입하여 혼합정수선형계획을 정식화하였다. 둘째, 깊이 우선 탐색법을 사용한 분기한계 알고리즘을 개발하였다. 이때 초기해는 choi (1989)가 개발한 15가지 휴리스틱 알고리즘 중 가장 우수한 알고리즘의 해를 사용하여 최적해 탐색에 대한 효율성을 제고시켰다. 개발된 분기한계 알고리즘의 성능을 평가하기 위하여 3가지 요인 즉, 가공시간과 준비시간의 비율(PS), 기계의 수(M) 및 작업의 수(N)로 실험설계하여 평균 노드 수를 검토하였다.

2. 假定과 符號

본 논문에서 연구하고자 하는 모형은 준비시간이 종속적인 다단계 흐름작업의 일정계획에서 작업 총 완성시간을 최소화하는 것으로 다음의 가정을 전제로 한다.

1. 기계의 수와 형태는 알고 있고 변하지 않는다.
 2. 첫번째 작업이 시작되는 시점에서 모든 기계는 이용 가능하다.
 3. 모든 기계는 항상 이용 가능하다. 즉, 고장, 보수, 교대작업 및 일간변동에 따른 변화가 없다.
 4. 각 작업은 4가지 상태 즉, 다음 기계에 대한 대기상태, 준비상태, 가공상태 및 가공 마무리 상태 중 하나이다.
 5. 어떠한 작업대상도 완료되기 전에는 시스템을 떠날 수 없다. 즉, 작업분할, 타부서 이관 및 외부하청이 불가능하다.
 6. 한 기계가 특정작업을 완료할 때까지는 다른 작업이 시작될 수 없다.
 7. 현 기계에 대한 준비는 전 기계에서 작업이 완료된 후 현 기계에 도착되어야 가능하다.
 8. 모든 작업에 대한 기계의 가공순서는 같으며 항상 고정되어 있다.
 9. 준비시간은 작업순서에 종속적이다.
 10. 모든 작업은 초기에 가공준비가 되어 있다.
 11. 다음 기계에 대한 작업의 이동시간은 무시한다.
 12. 각 작업에 대한 납기는 고려하지 않는다.
- 본 논문에서 사용되는 부호의 정의는 다음과 같다.

P : 가공시간

S : 준비시간

PS : 가공시간과 준비시간의 비율

N : 작업의 수

M : 기계의 수

r : 작업순서의 위치 ($r=1, 2, \dots, N$)

i : 작업j의 선행작업번호 ($i=1, 2, \dots, N$)

j : 작업번호 ($j=1, 2, \dots, N$)

K : 기계번호 ($k=1, 2, \dots, M$)

Z_{ir} : 1, 작업j가 작업순서 위치 r에 배정
0, 기타

MI : 기계대기

- JI : 작업대기
- C : 가장 늦은 완료시간
- E_M : 작업 총 완성시간
- I_{ik} : 1, 작업 i 가 기계 k 에 초기배정
0, 기타
- G_{ij} : 1, 작업 i 가 작업 j 에 대해 선행배정
0, 기타
- Q : 매우 큰 실수값
- BV : 이진변수의 수
- CV : 연속변수의 수
- ST : 제약조건식의 수
- BB : 하한계
- J : 작업군을 나타내는 부분 순열
- J' : 부분 순열의 보수
- K' : 기계 k 를 제외한 나머지 기계
- LEVEL : 수준 계수기

3. 整數計画法

M대 기계, N개 작업의 문제에 대한 결정변수를 행렬 Z와 열벡터 Z(r)로 표시하면 다음과 같다.

$$Z = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1N} \\ Z_{21} & Z_{22} & \cdots & Z_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{N1} & Z_{N2} & \cdots & Z_{NN} \end{bmatrix}$$

단, $Z_{jr} = 1$, 작업 j 가 작업순서 위치 r 에 배정
0, 기타

$$Z(r) = \begin{bmatrix} Z_{1r} \\ Z_{2r} \\ \vdots \\ Z_{Nr} \end{bmatrix}^T$$

단, 모든 r 에 대해서 성립

여기에서 추가로 사용된 변수는 기계대기(MI)와 작업대기(JI)로서 유휴시간을 나타낸다. 즉, MI_{rK} 는 r 번째 작업이 도착되지 않아 기계 K 가 대기하고 있는 유휴시간을 나타내며, JI_{rK} 는 $r-1$ 번째 작업이 k 단계에서 완료되지 않아 r 번째 작업이 대기하고 있는 유휴시간을 나타낸다.

가공시간은 행렬 P와 행벡터 P(k)로 표시하면 다음과 같다. 여기서 원소 P_{jk} 는 k 단계에서 작업 j 에 대한 가공시간을 나타낸다.

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1M} \\ P_{21} & P_{22} & \cdots & P_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NM} \end{bmatrix}$$

$$P(K) = \begin{bmatrix} P_{1K} \\ P_{2K} \\ \vdots \\ P_{NK} \end{bmatrix}^T$$

단, 모든 k에 대해서 성립

k단계에서 준비시간은 행렬 S_k 와 열벡터 $S_k(r)$ 로 표시하면 다음과 같다. 여기서 원소 S_{ijk} 는 k단계에서 선행 작업이 i일 경우 후속작업 j에 대한 준비시간을 나타낸다.

$$S_k = \begin{bmatrix} S_{11K} & S_{12K} & \cdots & S_{1NK} \\ S_{21K} & S_{22K} & \cdots & S_{2KK} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1K} & S_{N2K} & \cdots & S_{NNK} \end{bmatrix}$$

단, $S_{ijk}=0$, 모든 $i=j$ 에 대해서 성립

$$S_k(r) = \begin{bmatrix} S_{1jK} \\ S_{2jK} \\ \vdots \\ S_{NjK} \end{bmatrix}$$

단, $Z_{jr}=1$

작업 총 완성시간을 최소화하는 목적함수식은 식(1)과 같다.

$$\text{Min} \left[\sum_{k=1}^M P(K) Z(1) + \sum_{r=2}^N \{S_M(r) Z(r-1) + P(M) Z(r)\} + \sum_{r=2}^N M I_{rM} \right] \quad (1)$$

식(1)은 마지막 단계에서의 작업완료시간 즉, 작업 총 완성시간을 나타내는 식으로 첫번째 항은 준비시간이 필요없는 초기 작업이 M단계를 거쳐 가공되는 시간이다. 두번째 항은 초기 작업을 제외한 모든 작업이 마지막 단계에서 소요되는 준비 시간과 가공시간의 합이며, 세번째 항은 마지막 단계에서 초기작업을 제외한 모든 작업간의 기계 유휴시간을 나타낸다.

여기서 제약조건식은 7가지가 존재하는데 이것을 차례로 정식화하기로 한다.

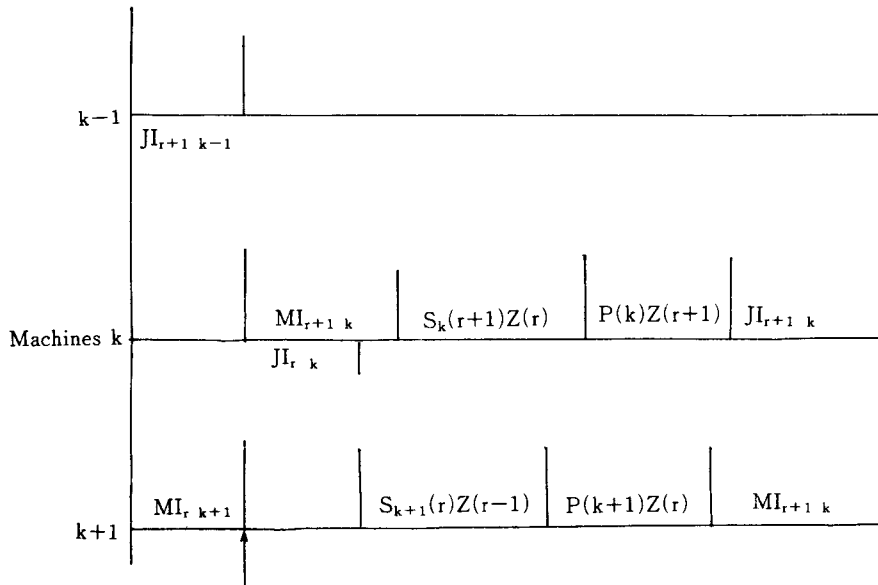
각 작업순서에 1개의 작업이 배정되는 제약식은 식(2)와 같다.

$$\sum_{j=1}^N Z_{jr}=1 \text{ 단, } r=1, 2, \dots, N \quad (2)$$

각 작업이 하나의 작업순서에 배정되는 제약식은 식(3)과 같다.

$$\sum_{r=1}^N Z_{ir}=1 \text{ 단, } j=1, 2, \dots, N \quad (3)$$

그림 1에서 단계K와 K+1사이의 관계를 제약식으로 나타내면 식(4)와 같다.



The moment job in position r is finished on machine k The moment job in position r+1 starts on machine k+1
Figure 1 Gantt Chart illustrating Integer Programming

$$MI_{r+1k} + S_k(r+1)Z(r) + P(k)Z(r+1) + JI_{r+1k} - JI_{rk} - S_{k+1}(r)Z(r-1) - P(k+1)Z(r) - MI_{L+1k+1} = 0 \quad (4)$$

단, $k=1, 2, \dots, M-1$
 $r=1, 2, \dots, N-1$

$r+1$ 번째 작업이 k 단계에서 완료된 후에 $k+1$ 단계에서는 3가지 상태 즉, (1) r 번째 작업이 $k+1$ 단계에서 완료된 상태 (2) $r+1$ 번째 작업이 도착되지 않아 기계 $k+1$ 이 대기하고 있는 상태 (3) r 번째 작업이 $k+1$ 단계에서 완료되지 않아 $r+1$ 번째 작업이 대기하고 있는 상태가 발생한다. 그러므로 기계대기와 작업대기가 동시에 발생하지 않는 제약식은 식(5)와 같다.

$$(MI_{r+1k+1}) \cdot (JI_{r+1k}) = 0 \quad (5)$$

단, $k=1, 2, \dots, M-1$
 $r=1, 2, \dots, N-1$

또한, 초기 단계에서의 유희시간(기계대기)와 첫번째 작업의 유희시간(작업대기)이 존재하지 않는 제약식은 식(6)과 같다.

$$MI_{r+11} = 0 \quad \text{단, } r=1, 2, \dots, N-1 \quad (6)$$

$$JI_{1k} = 0 \quad \text{단, } k=1, 2, \dots, M-1 \quad (7)$$

끝으로 첫번째 작업의 준비시간이 필요없는 제약식은 식(8)과 같다.

$$S_{k+1}(1)Z(0) = 0 \quad \text{단, } k=1, 2, \dots, M-1 \quad (8)$$

이상의 정수계획 모형에 대한 정식화를 요약하면 그림2의 식(1)부터 식(8)과 같다.

그러나 이 정식화로 실제 문제를 적용하게 되면 결정변수 Z 는 작업간의 준비시간으로 인하여 승법 형태로 나타나게 되어 해를 구하기가 어렵게 된다.

따라서 다음 장에서는 이 정수계획 모형을 선형으로 정식화할 수 있는 혼합정수선형계획에 대해 기술하기로 한다.

Objective function :

$$\text{Min} \left[\sum_{k=1}^M P(k)Z(1) + \sum_{r=2}^N \{S_M(r)Z(r-1) + P(M)Z(r)\} + \sum_{r=2}^N MI_{rM} \right] \quad (1)$$

Subject to :

$$\sum_{j=1}^N Z_{jr} = 1 \quad \text{for } r=1, 2, \dots, N \quad (2)$$

$$\sum_{r=1}^N Z_{jr} = 1 \quad \text{for } j=1, 2, \dots, N \quad (3)$$

$$MI_{r+1k} + S_k(r+1)Z(r) + P(k)Z(r+1) + JI_{r+1k} - JI_{rk} - S_{k+1}(r)Z(r-1) - P(k+1)Z(r) - MI_{L+1k+1} = 0 \quad (4)$$

for $k=1, 2, \dots, M-1$
 $r=1, 2, \dots, N-1$

$$(MI_{r+1k+1}) \cdot (JI_{r+1k}) = 0 \quad \text{for } k=1, 2, \dots, M-1$$

$$r=1, 2, \dots, N-1 \quad (5)$$

$$MI_{r+11} = 0 \quad \text{for } r=1, 2, \dots, N-1 \quad (6)$$

$$JI_{1k} = 0 \quad \text{for } k=1, 2, \dots, M-1 \quad (7)$$

$$S_{k+1}(1)Z(0) = 0 \quad \text{for } k=1, 2, \dots, M-1 \quad (8)$$

$$Z_{jr} = 0 \text{ or } 1$$

Figure 2. IP Formulation

4. 混合整數線型計劃法

본 연구에서는 모든 식을 선형으로 바꾸기 위해 3장에서 이진변수 Z_{jr} (작업 j 가 r 번째 배정되면 1이고 아니면 0)를 사용하지 않고 새로운 이진변수 G_{ij} (작업 i 가 작업 j 에 대해 선형배정되면 1이고 아니면 0)를 사용하게 된다.

작업 총 완성시간을 최소화하는 것은 결국 마지막 단계의 작업완료시간을 최소화하는 것과 같으므로 목적함수식 및 제약식은 식(9)와 식(10)과 같다.

$$\text{Min } E_M \tag{9}$$

$$E_M \geq C_{jM} \text{ 단, } j=1, 2, \dots, M \tag{10}$$

초기단계에서 임의의 작업 j 에 대한 완료시간이 적어도 가공시간 보다 커야하는 제약식은 식(11)과 같다.

$$C_{j1} \geq P_{j1} \text{ 단, } j=1, 2, \dots, N \tag{11}$$

각 단계에서 초기작업은 준비시간이 필요없으므로 이진변수 I_{jk} (작업자 j 가 k 단계에 초기배정되면 1이고 아니면 0)를 도입하면 된다. 이 변수를 이용하여 단계별 완료시간의 관계를 표시하면 식(12)와 식(13)과 같다.

$$C_{jk} \geq C_{j(k-1)} + P_{jk} + (1 - I_{jk})S_{ijk} \tag{12}$$

$$\text{단, } k=2, 3, \dots, M$$

$$i=1, 2, \dots, N$$

$$j=1, 2, \dots, N$$

$$\sum_{j=1}^N I_{jk} = 1 \text{ 단, } k=2, 3, \dots, M \tag{13}$$

끝으로 모든 단계에서 작업 i 와 작업 j 는 선행관계에 따라 식(14)와 식(15) 두 제약식 중 하나를 만족하게 된다. 여기서 Q 는 매우 큰 실수값을 나타낸다.

$$C_{ik} - C_{jk} + G_{ij} \geq S_{ijk} + P_{ik} \tag{14}$$

$$\text{단 } k=2, 3, \dots, M$$

$$i=1, 2, \dots, N$$

$$j=1, 2, \dots, N$$

$$j > i \geq 1$$

$$C_{jk} - C_{ik} + Q(1 - G_{ij}) \geq S_{ijk} + P_{jk} \tag{15}$$

$$\text{단, } k=2, 3, \dots, M$$

$$i=1, 2, \dots, N$$

$$j=1, 2, \dots, N$$

$$j > i \geq 1$$

이와 같은 두 식으로 결정되는 4개의 변수 중 I 와 G 는 이진변수이고 나머지 두 변수 E_M 과 C 는 연속변수에 해당한다.

이상의 혼합정수 선형계획에 대한 모형을 정식화하면 그림3의 식(9)부터 식(15)와 같다.

그러나 이 정식화 문제는 작업의 수(N)와 기계의 수(M)에 따라 차원이 달라지게 된다. 즉 이진 변수의 수(BV), 연속변수의 수(CV) 및 제약조건식의 수(ST)에 대한 차원은 N 과 M 에 의해 식(16), 식(17) 및 식(18)과 같이 표현된다.

$$IV = \frac{N(N-1)}{2} + (M-1)N \tag{16}$$

$$CV = MN + 1 \tag{17}$$

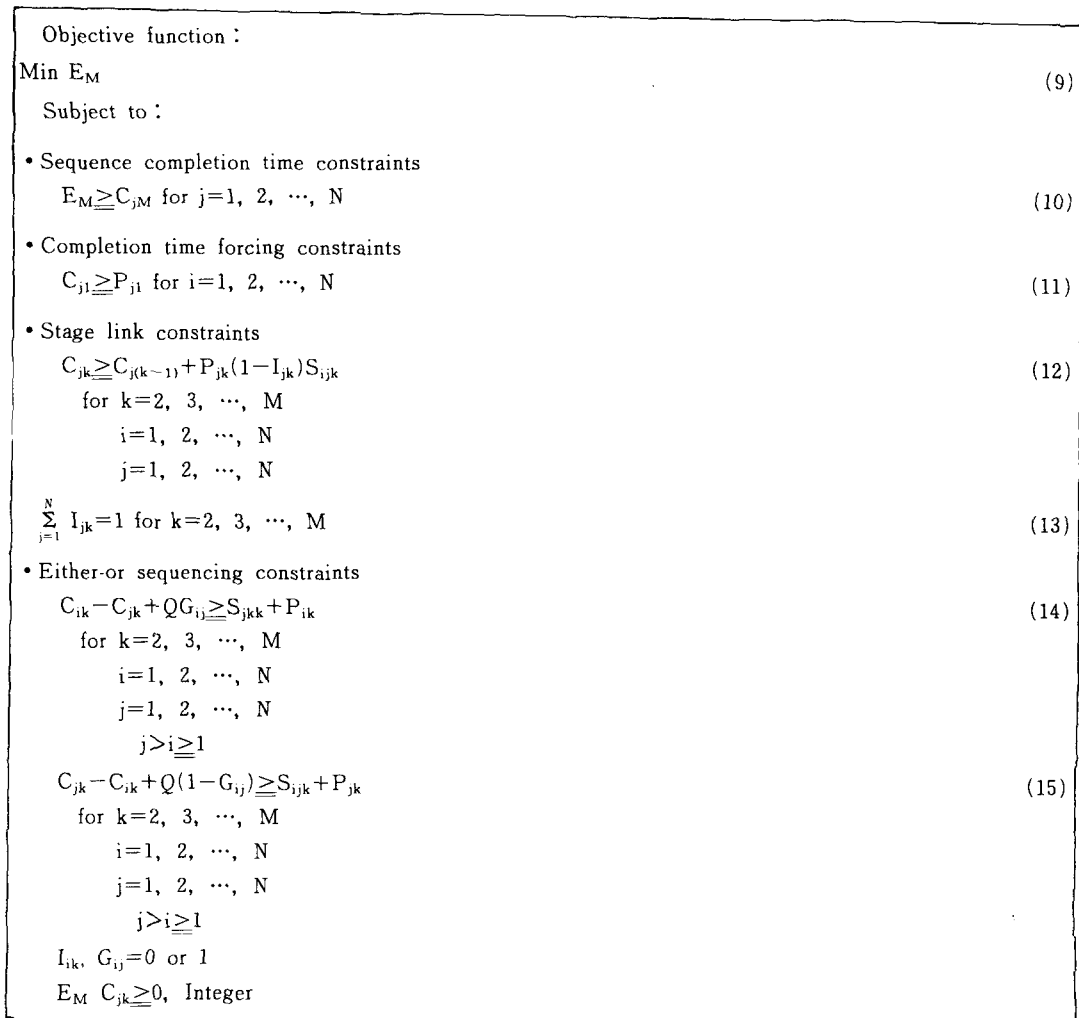


Figure 3 MILP Formulation

$ST = 2N + (N^2 + 1)(M - 1) + MN(N - 1)$ (18)

MILP로 최적해를 구하려면 표1과 같은 소규모 문제에서도 많은 변수와 제약식을 갖게 되므로 본 연구에서는 개선된 분기한계 알고리즘으로 최적해를 구하기로 한다.

Table 1 Problem Dimension of MILP

M \ N	4	8	10
5	IV* = 25 CV** = 21 ST*** = 168	IV = 45 CV = 41 ST = 352	IV = 55 CV = 51 ST = 444
6	IV = 33 CV = 25 ST = 243	IV = 57 CV = 49 ST = 511	IV = 69 CV = 61 ST = 645
7	IV = 42 CV = 29 ST = 332	IV = 70 CV = 57 ST = 700	IV = 71 CV = 84 ST = 884

*: number of binary variables
 **: number of continuous variables
 ***: number of constraints

5. 分枝限界 알고리즘

분기한계 알고리즘은 분기과정과 한계과정의 반복에 의해 이산적 조합문제에 대한 최적해를 얻는 묵시적 열거법으로 완전열거법(complete enumeration method)보다는 더 효율적이다.

분기과정에서 작업순서의 부분집합에 대응하는 노드(node)가 생성된다. 이때 가장 적은 반복 계산으로 최적의 노드를 찾기 위하여 각 노드에 대한 하한계(LB: Lower Bound)를 계산하여야 한다. 이러한 한계과정에 의하여 구한 값은 현 일정에 대한 예측값이며 나머지 작업의 순서와 관계없이 현재의 노드에 대한 작업 총 완성 시간으로서 필요한 값을 나타낸다. 이러한 하한계를 수식으로 나타내면 식(19), 식(20) 및 식(21)과 같으며 각 노드의 하한계는 식(21)을 이용하여 구한다.

$$BB(K) = C(J, K) + \sum_j (S_{JK} + P_{JK}) + \min_j \sum_{k=k+1}^M (S_{JK} + P_{JK}) \quad (19)$$

단, $1 \leq K \leq M-1$

$$BB(M) = C(J, M) + \sum_j (S_{JK} + P_{JK}) \quad (20)$$

$$BB = \max \{BB(K) \mid 1 \leq K \leq M\} \quad (21)$$

식(21)은 모든 단계에서의 하한계중 가장 큰 값이 그 노드에서의 완성시간으로 채택되는 것을 의미한다. 또한, 각 수준(level)에서의 하한계(LB)의 값이 큰 노드가 최적해에 좋지 않은 영향을 주므로 제거하며 값이 가장 작은 노드를 선택하여 분기과정을 수행한다.

분기과정에서 해를 탐색하는 두가지 방법 즉, 너비 우선 탐색법(Breadth-first-search, frontier-search, branch-from-lowest Search, Jumptracking method)과 깊이 우선 탐색법(depth-first-search, branch-from-newest-bound search, backtracking method)이 있다. 너비 우선 탐색법은 한 노드에서 하한계 값이 가장 작은 인접노드를 탐색하지만, 깊이 우선 탐색법은 한 경로 즉, 초기해를 찾은 다음 경로에서 역으로 추적하며 하한계 값을 비교하면서 탐색한다. 너비 우선 탐색법은 깊이 우선 탐색법 보다 계산량이 적어 빠르게 최적해를 구할 수 있으나 메모리 용량이 많이 소요된다는 단점이 있다. 따라서 본 연구에서는 프로그램의 효율성을 높이기 위해 깊이 우선 탐색법을 사용하기로 한다.

본 연구에서는 깊이 우선 탐색법으로 초기해를 구할 경우 Choi(1989)가 개발한 휴리스틱 알고리즘의 해를 사용하기로 한다. 이러한 분기한계 알고리즘을 단계별로 나타내면 다음과 같다.

단계 1 : (초기과정)

Choi(1989)가 개발한 알고리즘의 해에 해당하는 경로를 찾아 초기해를 구하며 이 경로의 하한계 BB^* 를 구한다. 수준계수기 $LEVEL=N$ 으로 놓고 단계 2로 간다.

단계 2 : (역진과정)

전수준으로 역진한다. 수준계수기 $LEVEL=LEVEL-1$ 로 놓는다.

(1) $LEVEL=1$ 이고 남은 노드들이 없으면 이때의 임시해가 최적해가 되면 남은 노드가 있으면 단계 4로 간다. 2) $LEVEL>1$ 이면 단계 3으로 간다.

단계 3 : (노드점도과정)

현 수준에서 분기된 노드가 남아 있지 않으면 단계 2로 가고 분기될 노드가 남아 있으면 단계 4로 간다.

단계 4 : (하한계 비교과정)

임시해 BB^* 와 각 노드의 하한계 BB 를 비교한다. 이때 하한계는 식(21)을 이용하여 구한다.

1) $BB^* > BB$ 이고 $LEVEL=N$ 이면 $BB^*=BB$ 로 놓고 $LEVEL \neq N$ 이면 새로운 해를 탐색하기 위해 진진(분기)과정으로 들어간다. 수준계수기 $LEVEL=LEVEL+1$ 로 놓는다. 단계 4로 간다. 2) $BB^* \leq BB$ 이면 해당 노드를 제거시키고 단계 2로 간다.

분기한계 알고리즘의 계산 효율성을 알아보기 위해 3가지 요인의 각 수준에서 하한계를 비교하기 위한 노드 수를 검토하였다.

표2에서 비교된 평균 노드 수를 보면 116개이고, 작업수가 커지면 노드 수는 증가한다(표 5). 그러나 가공시간과 준비시간의 비율, 기계의 수는 노드의 수에 별로 영향을 미치지 않는다(표 3, 표 4). $PS \times M = 0.5 \times 4$ 수준에서 95개(표 6), $PS \times N = 0.5 \times 4$ 수준에서 44개(표 7), $M \times N = 4 \times 5$ 수준에서 57개(표 8), $PS \times M \times N = 0.5 \times 4 \times 5$ 수준에서 40개(표 9) 노드 수가 가장 적게 나타났음을 알 수 있다.

Table 2 Number of Nodes

Mean	Median	Minimum	Maximum
116(360*)	114	40	192

* : Number of runs

Table 3 Mean Number of Nodes : PS

PS	0.5	1.0	1.5	2.0
	116	118	119	118

* | Number of runs

Table 4 Mean Number of Nodes : M

M	4	8	10
	114(120*)	117	115

* : Number of runs

Table 5 Mean Number of Nodes : N

M	5	6	7
	58(120*)	111	115

* : Number of runs

Table 6 Mean Number of Nodes : PSXM

M \ PS	0.5	1.0	1.5	2.0
4	95(30*)	120	121	118
8	116	117	119	117
10	106	117	117	118

* : Number of runs

Table 7 Mean Number of Nodes : PSXN

M \ PS	0.5	1.0	1.5	2.0
5	44(30*)	62	63	62
6	110	112	109	112
7	163	180	117	179

* : Number of runs

Table 8 Mean Number of Nodes : MXN

N \ M	4	8	10
5	57(40*)	57	59
6	109	112	112
7	175	182	173

* : Number of runs

Table 9 Mean Number of Nodes : PSXMXN

PS \ M		0.5			1.0			1.5			2.0		
		4	8	10	4	8	10	4	8	10	4	8	10
N	5	40(10*)	42	50	62	62	62	64	62	61	62	62	62
	6	100	115	114	116	110	110	105	110	112	114	110	110
	7	145	189	154	182	179	177	192	183	177	178	177	181

* : Number of runs

6. 결론

본 연구에서는 준비시간이 작업순서에 종속적인 다단계 흐름작업의 일정계획에 대한 최적 알고리즘을 개발하였다. 다단계 SDST흐름작업의 일정계획에서는 준비시간이 작업순서에 따라 크게 변할 수 있으므로 전 단계에서 현 단계에 도착된 작업의 준비시간이 작업 총 완성시간에 영향을 크게 미친다. 따라서 본 논문에서는 기존연구와 달리 현 단계에서의 준비는 전 단계에서 작업이 도착된 후에 수행될 수 있는 것으로 가정하였다.

본 연구에서 얻은 결과를 다음 2가지로 나누어서 요약·수정한다.

1. 본 연구에서 제시한 SDST흐름작업의 일정계획모형에 대해 정수계획으로 정식화하였으며 그리고 새로운 이진변수를 도입하여 혼합정수계획을 정식화하였다.

2. 길이 우선 탐색법을 사용한 분기한계 알고리즘을 개발하였다. 이때 초기해는 Choi(1989)가 개발한 휴리스틱 알고리즘의 해를 이용하여 최적해 탐색법에 대한 효율성을 제고시켰다. 개발된 휴리스틱 알고리즘의 성능을 평가하기 위하여 3가지 요인 즉, 가공시간과 준비시간의 비율, 기계의 수 및 작업의 수로 실험설계하여 평균 노드 수를 검토하였다.

참고문헌

1. Choi, S. W., "Analysis of Multi-stage Flowshop Scheduling with Sequence Dependent Setup Time," Ph. D. Dissertation, Industrial Engineering Department, Hanyang University, 1989.
2. Gupta, J. N. D., "n Jobs and M Machines Job-Shop Problems with Sequence-Dependent Set-Up Times," Int. J. Prod. Res., Vol. 20, No. 5, pp. 643-56, 1982.
3. _____, "Flowshop Schedule with Sequence Dependent Setup Times," Journal of Operations Research Society of Japan, Vol. 29, No. 3, pp. 206-19, 1986.
4. _____, and W. P., Darrow, "The Two-Machine Sequence Dependent Flowshop Scheduling Problem," European Journal of Operational Research, Vol. 24, No. 3, pp. 439-46, 1986.
5. Ignall, E., and L., Schrage, "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, Vol. 13, No. 3, pp. 402-12, 1965.
6. Lomnicki, Z. A., "A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," Opl. Res. Q., Vol. 16, No. 1, pp. 89-100, 1965.
7. Srikar, B. N., and S Ghosh, "A MILP Model for the n-Job, m-Stage Flowshop with Sequence Dependent Set-Up Times," Int. J. Prod. Res., Vol. 24, No. 6, pp. 1459-74, 1986.
8. Stafford, E. F., "On the Development of a Mixed-Integer Linear Programming Model for the Flowshop Sequencing Program," J. Opl. Res. Soc., Vol. 39, No. 12, pp. 1163-74, 1988.
9. Szwarc, W., and Gupta, J. N. D., "A Flow-Shop Problem with Sequence-Dependent Additive Setup Times," Naval Research Logistics Quarterly, Vol. 34, No. 5, pp. 619-27, 1987.
10. Uskup, E., and S. B. Smith, "A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems," Operations Research, Vol. 23, No. 1, pp. 118-36, 1975.