Journal of the Korean Institute
of Industrial Engineers
Vol.24, N0.3, September, 1998

순서 의존적인 작업시간을 갖는 작업들의 스케쥴링을 위한 동적계획법*

Dynamic Programming Algorithms for Scheduling Jobs with Sequence-Dependant Processing Times

이문규** · 이승주**

Moon-Kyu Lee** · Seung-Joo Lee**

Abstract

In this paper, we consider the problem of scheduling n jobs with sequence-dependent processing times on a set of parallel-identical machines. The processing time of each job consists of a pure processing time and a sequence-dependent setup time. The objective is to maximize the total remaining machine available time which can be used for other tasks. For the problem, we first propose a dynamic programming(DP) algorithm for sequencing jobs processed on a single machine. The algorithm is then extended to handle jobs on parallel-identical machines. Finally, we developed an improved version of the algorithm which generates optimal solutions using much smaller amount of memory space and computing time. Computational results are provided to illustrate the performance of the DP algorithms.

1. 서 론

오늘날 중소기업이나 대기업을 막론하고 대부분의 제조업체는 다품종 소량생산 체제를 취하고 있다. 이러한 다품종 소량생산 체제하에서는 각 작업장의 기계들이 다양한 부품을 가공 또는 조립하기 위한 다기능을 갖추고 있다. 그런데, 각 작업의 소요시간은 대부분의 경우 前작업에 따라 준비시간이 변화하기 때문에 준비시간을 포함한 전체 작업시간은 작업순서에 따라 의존적인 경우가 대부분이다. 예로서, 작업마다 사용하는 공구가 다르면 직전 작업에 사용되던 공구들을 제거하고 새로운 공구들을 장착해야 하는 경우에나 동일한 공구를 사용하더라도 원자재가 달라져 이에 따른 준비시간이 필요한 경우를 들 수 있다. 본 연구에서는 이와 같이 소위 작업순서에 의존적인 작업시간(sequence-dependant processing time)을 갖는 작업들의 병렬기계 스케쥴링 문제(Parallel-machine Scheduling Problem with Sequence-dependent Processing Times: PSPSPT)를 다룬다. 이러한 스케쥴링 문제는 기계가공업체는 물론이고 섬유업체, 플라스틱, 성형가공업체, 석유화학업체 등의 작업현장에서 흔히 대두되는 문제이다[5, 7, 15].

순서에 무관한 작업들을 병렬기계에서 처리하는 일반

* 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

** 계명대학교 산업공학과

적인 문제에 대한 기존의 연구는 Cheng & Sin[4]에 정리된 바와 같이 그 연구실적이 상당하다. PSPSPT에 관한 기존의 연구도 어느 정도 진행되어 왔는데 혼합정수계획으로 최적화모형을 정립한 후 선형계획법과 rounding을 이용하여 근사최적해를 구하는 시도가 있었고[5, 16], 최근에는 2종류의 작업이 있을 경우 최적해를 polynomial 시간내에 구할 수 있는 알고리즘을 제시한 연구[14]와 총 완료시간을 최소화하기 위한 동적계획법을 제안한 연구[3] 등이 발표되었다. 또한, 이 문제에 대한 발견적해법을 제시한 연구가 발표된 바 있다[8, 10, 12, 13].

이상의 기존 연구들에서 고려되고 있는 목적함수는 작업완료시간의 최대값(C_{max})이나 총 준비시간(total setup time)을 최소화하거나 납기가 주어진 경우 지연시간(lateness time)의 최대값(L_{max})을 최소화하는 경우가 대부분이다. 그러나 실제 생산현장에서는 이러한 목적함수와는 달리, 작업할당 후 남는 잔여기계 가용시간(remaining machine available time: RMAT)을 최대화하는 목적함수를 고려하는 것이 더 바람직할 경우가 많다. 이는, 주어진 업무시간내에 모든 작업들을 가능한 최대로 각 기계에서 수행할 때, 작업이 전혀 할당되지 않은 기계의 가용시간과, 할당된 기계들 중 완료시간이 최소인 기계의 남은 가용시간의 합을 최대화하여 이 시간을 다른 생산작업에 활용하고자 하는 현장의 요구를 반영한 것이다.

이러한 상황을 나타내는 대표적인 작업장으로는 전기·전자적인 신호를 전달하기 위하여 자동차나 컴퓨터에서 필수적으로 사용되는 와이어하니스(wire harness)의 주요 부품인 압착단자(crimped terminal)를 병렬의 동일기계에서 가공하는 작업장을 들 수 있다[1, 2]. 여기서는 압착단자의 주문량이 일정치 않아서 설치된 기계들을 모두 사용하지 않는 경우가 종종 있는데, 이 경우 작업이 할당되지 않은 기계는 다른 종류의 작업에 활용될 수 있다. 그러므로 주어진 작업들을 처리하고 남는 가용시간을 최대화하여 그 활용도를 높이는 것이 중요한 과제가 된다. 특히 작업수가 많을 때는 가동해야 할 기계대수를 사전에 정확하게 예측하지 못하기 때문에 이러한 목적함수가 효과적이라고 생각된다.

기존 문헌에서 가장 많이 다루어진 C_{max}를 최소화하는 문제는 가동해야 할 기계대수가 일반적으로 미리 정해져 있어야 한다. 또한, C_{max}가 작업현장의 최대 가용시간(예로서, 일당 8시간)을 초과하는 경우에는 실제 실현 가능한(feasible) 해가 못되고, 반대로 C_{max}가 가용시간에 비해 너무 작게되면 RMAT를 최대화하여 가동기계대수를 가능한 최소화하려는 방법보다 각 기계에 남게 되는 시간에 대한 처리가 훨씬 비효율적일 것이다.

이상의 여러 가지 검토사항들을 감안하여 보면 RMAT를 최대화하는 문제가 실제 산업현장의 스케쥴링 분야에서 자주 대두되는 중요한 문제이고 또한 시급히 해결되어야 할 과제중의 하나라고 판단된다. 그러나, 이와 같이 RMAT를 최대화하려는 문제의 중요성에도 불구하고 기존의 순서의존적인 작업스케쥴링에 관한 연구분야에서는 아직까지 다루어진 바 없는 것으로 보인다. 본 연구에서는 이와 같이 생산현장의 보다 현실적인 상황을 반영하여 작업순서 의존적인 작업시간을 갖는 작업들을 RMAT가 최대화되도록 스케쥴링하는 문제인 PSPSPT를 위한 최적해법으로서 동적계획 모형(DP model)들을 제시하고자 한다.

2. 수리적 모형

본 연구에서는 다음의 용어를 사용한다:

J_i = 작업 i, i = 1, ⋯, n;

M_k = 기계 k, k = 1, ⋯, m;

P_j = 작업 j의 순수 처리시간;

S_{ij} = 작업 i가 완료된 후 작업 j를 처리하기 위한 작업준비시간
(i=1, ..., n; j=1, ..., n);

$S_{n+k,j}$ = 기계 k에서 작업 j가 최초작업으로서 처리될 때 발생하는 준비시간으로서 기계에 무관하게 모두 SM_j로 동일하다고 가정함
(k=1, ..., m; j=1, ..., n);

t_{ij} = 직전 작업이 i일 때 작업 j의 처리시간 = S_{ij} + P_j;

$C_M(k)$ = 기계 k의 완료시간;

MAT = 계획기간, 기계가용기간(machine available time) 또는 작업가능시간.

본 연구 대상의 문제는

"그 순서에 따라 작업 시간이 달라지는 n개의 작업들

을 주어진 계획기간내에 병렬로 구성된 m개의 동일기계에서 처리하기 위하여, 타 작업에 활용될 수 있는 기계 가용시간인 RMAT를 최대화하도록 각 기계에서 처리될 작업들과 그 처리순서를 결정하는 것"

으로 정의된다. 여기서, 계획기간은 각 기계가 동시에 작업을 처리한다는 가정하에 각 기계가 작업할 수 있는 총 가용시간을 의미한다. 최대화 대상의 목적함수 RMAT는 그림 1에 도시된 예와 같이 할당되지 않은 기계들의 MAT 값과, 가장 빨리 완료되는 기계 3의 완료시간과 MAT와의 절대차의 합으로 정의된다. 즉, RMAT만큼 다른 유형의 작업에 활용될 수 있기 때문에 이를 최대화하고자 하는 것이다.

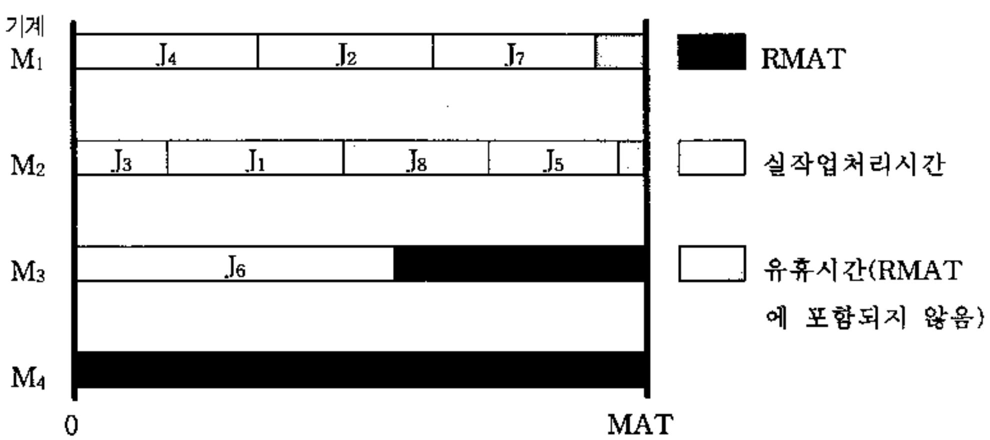

그림 1. 4대의 동일기계 (M_1, ⋯, M_4)에서 8개의 작업(J_1, ⋯, J_8)을 처리하기 위한 작업 스케줄링의 예

이 문제를 문제유형 정의를 위한 용어대로 표현하면 P/*sds*/-g(σ)가 되는데[8], g(σ)는 주어진 스케줄 σ에 대한 RMAT로서 작업이 할당되지 않은 기계들의 집합

$$U_m = \{M_k \mid C_M(k) = 0,\ k=1, \ldots, m\}$$

라고 할 때

$$g(\sigma) = MAT \cdot |U_m| + \max_{k \notin U_m}[MAT - C_M(k)]$$

와 같이 정의된다. 여기서 $|S|$는 집합 S의 원소수를 의미한다. 이 문제를 최적해가 보장되는 수리적 모형으로 정형화하면 다음과 같이 혼합정수계획(mixed linear integer programming)이 된다 :

$$\max \quad MAT \cdot |U_M| + \max_{k \notin U_m}[MAT - C_M(k)] \tag{1}$$

s.t.

$$\sum_{i=0, i\neq j}^{n} \sum_{k=1}^{m} x_{ijk} = 1 \quad j = 1, \cdots, n \tag{2}$$

$$\sum_{i=0, i\neq h}^{n} x_{ihk} - \sum_{j=0, j\neq h}^{n} x_{hjk} = 0$$
$$h = 1, \cdots, n;\ k=1, \cdots, m \tag{3}$$

$$u_{ik} - u_{jk} + (n+1)x_{ijk} \le n$$
$$0 \le i \neq j \le n;\ k = 1, \cdots, m \tag{4}$$

$$C_M(k) = \sum_{j=1}^{n}(S_{n+k,j} + P_j)x_{0jk} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} t_{ij}\, x_{ijk} \le MAT$$
$$k = 1, \cdots, m \tag{5}$$

$x_{ijk} \in \{0, 1\}$, I=0, 1,⋯, n-1; j =0, 1, ⋯, n; k = 1, ⋯, m.

$u_{ik} \ge 0$ (정수) i = 0, ⋯, n-1; k = 1, ⋯, m.

여기서 결정변수 x_{ijk}는 작업 j가 작업 i 바로 다음에 기계 k에서 처리될 때는 1이고 그렇지 않으면 0으로 정의된다(단, 작업 0는 각 기계마다 가상적으로 정의되는 최초의 작업을 의미하므로 x_{0jk}는 작업 j가 기계 k에서 최초 작업으로 처리될 때는 1이고 그렇지 않으면 0으로 정의되고 x_{i0k}는 작업 i가 기계 k에서 마지막 작업으로 처리될 때는 1이고 그렇지 않으면 0을 의미함). u_{ik}는 subtour를 방지하기 위하여 주어지는 조건식 (4)에 사용되는 실수변수이다. 목적함수 (1)은 RMAT이고 제약식 (2)와 (3)는 각각 작업 j가 오직 하나의 위치에 할당되도록 하는 제약조건과 흐름량보존(flow conservation) 조건을 나타낸다. 제약식 (5)는 각 기계의 가용용량 초과를 방지하기 위한 조건이다.

3. 동적계획 모형

이 문제는 복수 외판원문제(Multiple Traveling Salesmen Problem: MTSP)와 일면 유사한 점이 있으나 목적함수가 다르기 때문에 MTSP를 위하여 개발된 해법[9]을 그대로 사용할 수가 없다. 즉, MTSP에서는 복수외판원에 의해 발생되는 총 경로비용을 최소화하고자 하는데, 이것은 그림 1에 도시된 유휴시간까지 포함된 총 RMAT를 최대화하는 것으로 해석할 수 있다. 이 결과 다른 생산작업에 활용되기 어려운 유휴시간을 극대화하게 하여 결과적으로는 필요 이상의 기계를 가동해야 할 경우가 발생될 수 있다. 즉, 극단적인 경우 그림 1의 예에서 기계 4까지도 작업이 할당될 가능성이 있게 된다는 것이다. 이 경우에는 모든 기계들이 불필요하게 상당한 유휴시간을 갖게 되어 기계별로 남는 시간을 효과적으로 이용할 수가 없게 된다. 이것 보다는 기계 가동효율과 생산성 측면에서 볼 때 유휴시간을 소수의 기계에 집중화시키는 것이 훨씬 바람직할 것이다

이상과 같이 정형화된 문제의 최적해를 구하기 위하여 본 연구에서는 외판원문제나 단일기계의 작업순서화(sequencing) 문제를 다룬 연구[11]와, 작업순서와 무관한 작업시간을 갖는 작업들을 병렬기계에서 처리하기 위한 스케쥴링 해법으로서 동적계획법을 활용한 연구[3, 6] 등을 참고하여 동적계획법에 기초한 최적해법을 개발하였다.

3.1 단일기계를 위한 동적계획법(Dynamic Programming for Single-Machine Scheduling: DPSS)

여기서는 기계가 1대일 경우로서 Held & Karp[11]의 연구를 기초로 하여 본 문제의 특성에 적합하도록 다음과 같이 수정하였다. 사용되는 용어로서,

$J = \{J_1, J_2, \cdots, J_n\}$;

R = 기계에 이미 할당된 작업들의 집합;

$l = |R|$;

$G_q(R)$ = 집합 R의 모든 원소를 마지막 작업이 $J_q \in R$인 순서로 가공했을 때의 최대 RMAT;

M = 임의의 양의 큰 상수;

$$f_1(x) = \begin{bmatrix} -M, & \text{if } x < 0 \\ x, & \text{otherwise;} \end{bmatrix}$$

$$f_2(x) = \begin{bmatrix} M, & \text{if } x < 0 \\ x, & \text{otherwise} \end{bmatrix}$$

라고 정의한다. 여기서, 기계에 이미 그 순서가 할당된 작업수 l을 stage로, 할당된 작업집합 R을 state로 생각하면 순환식(recursive equation)은 다음과 같다:

우선

$$E_l(R, J_q) = \begin{bmatrix} MAT - (SM_q + P_q), & l = 1 \\ \max_i (G_i(R - \{J_q\}) - (S_{iq} + P_q)), & l > 1 \end{bmatrix}$$

으로 정의하면

$$G_q(R) = f_1(E_l(R, J_q));$$
$$G^*(R) = f_2(\max_q [G_q(R)]), \quad R \subseteq J;$$

이 된다. 여기서 R = J 일 때 최적 RMAT 값은 $G^*(R=J)$인데 그 값이 M과 같거나 클 때는 가능해가 없는 것으로 판정한다. $G^*(J)$값이 M보다 작을 경우 $J_{[i]}$를 i번째 처리하는 작업이라고 할 때 최적해 $R^* = \{J_{[1]}, J_{[2]}, \cdots, J_{[m]}\}$은 다음 식을 순차적으로 적용하여 구할 수 있다:

$$G^*(\{J_{[1]}, J_{[2]}, \cdots, J_{[i]}\}) = G^*(\{J_{[1]}, J_{[2]}, \cdots J_{[i-1]}\}) - t_{[i-1],[i]},$$
$$1 \leq i \leq m$$

3.2 복수기계에서의 동적계획법(Dynamic Programming for Multi-machine Scheduling: DPMS)

복수기계의 스케쥴링을 위한 동적계획법은 고려대상 기계를 stage로 하고, 각 기계에 할당되는 작업집합을 state로 생각하여 앞절의 단일기계 모형을 확장한 것이다. 여기서

R_k = k번째 기계에 할당된 작업들의 집합 ;

Q_k = 첫 번째 기계에서 k번째 기계까지 할당된 작업들의 집합, $Q_k \subseteq J$;

$= \cup_{j=1}^{k} R_j$;

$l_k = |R_k|$;

$F_k(Q_k)$ = Q_k의 작업들 중 기계 k에 R_k가 할당되었을 때의 최대 RMAT;

$$f_3(x) = \begin{cases} -M, & \text{if} \quad x \ge M \\ x, & \text{otherwise} \end{cases}$$

라고 정의하자. 그러면 k = 1일 경우의 순환식은 단일기계에서와 동일하므로

$$F_1(Q_1 = R_1) = G^*(R_1), \quad R_1 \subseteq J$$

가 된다. k ≥ 2일 때는

$$F_k(Q_k) = f_2(\max_{R_k} \quad f_3(E_2(R_k)));$$

$$E_2(R_k) = \begin{cases} \max[G^*(R_k), F_{k-1}(Q_k - R_k) - MAT \cdot |U_{k-1}|] \\ \quad + MAT \cdot |U_k|, & \text{if } R_k \notin \emptyset \\ MAT + F_{k-1}(Q_k), & \text{otherwise} \end{cases}$$

으로 표현된다. 여기서R_k^*를 $f_3(E_2(R_k))$를 최대화하는 집합으로 정의하자. 그러면 RMAT의 최대값은$F_m(Q_m = J)$가 되며, 최적해 R_k^*, k=1,...,m은 다음 식을 이용하여 k=m부터 k=1일 때 까지 역순으로 구하면 된다:

$$Q_m^* = J;$$

$$F_k(Q_k^*) = \begin{cases} \max[G^*(R_k^*), F_{k-1}(Q_k^* - R_k^*) - MAT \cdot |U_{k-1}|] \\ \quad + MAT \cdot |U_k|, & \text{if } R_k^* \notin \emptyset \\ MAT + F_{k-1}(Q_k^*), & \text{otherwise} \end{cases}$$

$$Q_{k-1}^* = Q_k^* - R_k^*, \quad 2 \le k \le m.$$

부록에서는 이상의 복수기계의 스케쥴링을 위한 동적계획법을 적용한 예를 보여주고 있다. 이 예제로부터 DPMS는 n과 m값의 증가에 따라 기하급수적으로 늘어나는 기억 용량($m\sum_{k=1}^{n} k \cdot {}_nC_k = m \cdot n \cdot 2^{n-1}$)이 필요하고, 또한 최종해에 있어서 동일한 해가 서로 다른 형태로 중복되게 나타나는 단점을 발견할 수 있다. 다음 절에서는 이러한 단점을 극복하기 위하여 본 연구에서 제안한 개선된 동적계획법을 설명한다. 개선된 동적계획법은 DPMS의 적용 과정에서 도출된 다음의 특성들을 바탕으로 하고 있다:

정리 I. PSPSPT의 최적해가$R_1^*, \cdots, R_u^*, \cdots, R_v^*, \cdots, R_m^*$일 때, 임의의 한쌍의 ($R_u^*$, R_v^*),($1 \le u \le v \le m$)의 순서를 서로 바꾼 해인$R_1^*, \cdots, R_v^*, \cdots, R_u^*, \cdots, R_m^*$도 역시 최적해이다.

((증명)) 기계 k에서 작업 j가 최초 작업으로 처리될 때 발생하는 준비시간이 기계에 무관하게 동일함으로, 모든 R_k의 단일기계에서의 최적 RMAT값은 기계에 무관하게 일정한 값을 가진다. 따라서 최적해R_k^*의 기계배치 순서를 바꾸더라도 최대 RMAT값은 변화하지 않는다.

정리 II. DPMS에 의하여 최적해를 구할 때, 만약 p번째 기계에서

$$F_p(Q_p^* = J) \ge 0 \tag{6}$$

을 만족하는 Q_p^*이 존재한다면, RMAT의 최대값은

$$F_p(J) + (m-p) \cdot MAT \quad (7)$$

가 된다.

((증명)) (6)식을 만족하는 Q_p*이 존재한다는 것은 모든 작업이 p대의 기계로 처리될 수 있다는 것이다. 즉, 정리 I에 따라 작업이 할당된 기계들을 기계 1부터 올림차순으로 정렬을 하였을 때, 기계 p 이후의 기계에는 할당된 작업이 없다는 것이다. 따라서, 기계 p이후의 기계에 어떠한 작업이라도 할당된다면, 그 스케쥴의 최대 RMAT는 기계 p까지의 최대 RMAT값에 기계 p 이후의 모든 기계의 MAT값을 더한 것보다 작아진다. 그러므로 기계 p까지의 최대 RMAT값은 (7)식으로 표현된다.

정리 Ⅲ. 기계가 1대인 경우, 모든 R ($|R| = l$, $0 \le l \le n$) 에 대하여 처음으로 G*(R) = M이 되는 l을 l*라 하고, $\overline{N}$를 1대의 기계에 최대로 할당 가능한 작업의 수라 하면,

$$\overline{N} = l^* - 1$$

이 되고, 원문제의 최적해를 $R_1^*, R_2^*, \cdots, R_m^*$ 이라고 할 때,

$$|R_k^*| \le \overline{N}, \quad 1 \le k \le n$$

이 성립한다.

((증명)) 만약 임의의 k에 대하여 $|R_k^*| > \overline{N}$ 라 하자. 그러면, R_k^*는 1대의 기계에 할당할 수 없는 작업들의 집합이 되어 $G^*(R_k^*) = M$ 이 된다. 이는 최적해의 가능해 특성에 위배되므로, 원문제의 최적해가 $R_1^*, R_2^*, \cdots, R_m^*$ 일 때, $|R_k^*| \le \overline{N}$, $1 \le k \le n$를 만족하여야 한다.

3.3 개선된 DPMS(Improved DPMS: IDPMS)

개선된 DPMS는 앞절에 요약된 정리들을 바탕으로 하여 개발되었다. 즉, 1대의 기계에 가능한한 최대로 작업들을 할당하여 기계수를 최소화해야 목적함수인 RMAT를 최대화할 수 있게 된다. 따라서, N_i를 기계 i에 할당되는 작업수이고 $\overline{m} = \mathrm{int}[n / \overline{N}]$ (여기서 int[a]는 실수 a보다 같거나 큰 최소의 정수를 의미함)라고 정의할 때 다음과 같은 할당 구조를 갖는 가능해 중의 하나가 최적해가 된다(여기서 할당 구조 $\Gamma(w)$는 총 w대의 기계로 주어진 작업들을 모두 처리할 때 각 기계에 할당된 작업수들의 집합을 나타냄):

$$\Gamma(\overline{m}) = \{N_1, N_2, \cdots, N_{\overline{m}}\},$$

$$\text{단, } N_k \le N_{k-1},\ k = 2, 3, \cdots, \overline{m} \text{ 이고 } \sum_{k=1}^{\overline{m}} N_k = n. \quad (8)$$

이렇게 주어지는 하나의 할당구조에 대한 소요 기억용량은

$$\sum_{k=1}^{\overline{m}+1} {}_nC_{N_k} \cdot N_k$$

으로서 $\overline{m}$가 주어졌을 경우 (8)과 같이 주어지는 모든 할당 구조를 고려한다 하더라도 앞의 DPMS의 경우와 비교하여 볼 때 미미한 수치이다. 이러한 할당 구조의 하나의 예는 마지막 기계를 제외한 모든 기계에 작업들을 최대로 할당하는 경우로서

$$N_k = \overline{N},\ \ k = 1, \cdots, \overline{m}-1,$$
$$N_{\overline{m}} = n - \overline{N}(\overline{m} - 1)$$

가 된다.

만약에 (8)의 할당구조를 갖는 모든 경우를 고려하였을 때 가능해가 나오지 않으면 그 다음으로 고려해야 할 할당구조는 한 대의 기계가 추가된 것으로서

$$\Gamma(\overline{m}+1) = \{N_1, N_2, \cdots, N_{\overline{m}}, N_{\overline{m}+1}\},$$

$$\text{단, } N_k \ge N_{k-1},\ \ k = 2, 3, \cdots, \overline{m}+1 \text{이고 } \sum_{k=1}^{\overline{m}+1} N_k = n \quad (9)$$

가 된다. 여기서도 가능해가 없으면 마찬가지 방법으로 한 대씩의 기계가 추가된 할당구조를 순차적으로 고려한다. 이러한 개념하의 IDPMS를 단계적으로 기술하면 다음과 같다:

STEP 1. 첫 번째 기계를 처리하는 경우, 즉 k = 1일 경우로서 이 때는 단일기계에서와 동일하므로 DPSS를 수행한다. 즉, $F_1(Q_1=R_1)=G^*(R_1)$, $\forall R_1 \subseteq J$ 를 계산한다.

if $G^*(R_1=J) \geq 0$,
{ RMAT의 최대값 = $G^*(R_1=J)$ + (m-1) · MAT;
최적해 $R_1^* = J$, $R_k^* = \phi$, $2 \leq k \leq m$;
stop.}
else
{ 모든 R_1에 대하여 $G^*(R_1) = M$이 되는 최소의 정수가 l^*이므로
$\overline{N} = l^* - 1$;
go to STEP 2.}

STEP 2. 할당 대상의 최대 기계대수가 w(초기치는 $\overline{m}$) 일 때의 모든 할당 구조

$\Gamma(w) = \{N_1, N_2, \cdots, N_w\}$ (단, $N_k \geq N_{k-1}$, k = 2, 3, ⋯, w 이고 $\sum_{k=1}^{w} N_k = n$)
를 생성한다.

STEP 3. STEP 2에서 생성된 모든 할당 구조에 대한 각각의 최대 RMAT를 다음과 같이 구한다:

for(k=1; k≤w ; k=k+1)
{$F_k(Q_k) = \max_{R_k} [f_3(E_2(R_k))]$;
$E_2(R_k) = \max[G^*(R_k), F_{k-1}(Q_k-R_k) - MAT \cdot |U_{k-1}|]$
+ MAT · $|U_k|$, $\overline{N} \geq l_1 \geq l_2 \geq \cdots \geq l_w \geq 1$.
}
if $F_w(Q_w) \geq 0$,
{ RMAT의 최대값 = $F_k(Q_k)$ + (m - w) · MAT;
stop.
}
else
{ w = w + 1;
repeat STEP 2 & 3.
}

3.4 IDPMS 적용 예제

여기서는 앞의 DPMS에 적용하였던 예제에 IDPMS를 재적용하여 풀어 보기로 한다.

STEP 1. 단계 1은 DPMS와 동일하게 수행되며, 부록의 k = 1인 경우로부터 4개 이상의 작업들이 하나의 기계에서 수행될 수 없는 것으로 나타났기 때문에 l^* = 4이고 따라서

$\overline{N} = l^* - 1 = 3$

이 된다.

STEP 2. n=5이므로 STEP 1의 결과로부터

$\overline{m} = 2$, $l_1 = |R_1| = 3$, $l_2 = |R_2| = 2$

라는 것을 알 수 있고, 이로부터 가능한 할당구조는 다음과 같이 생성된다:

$\Gamma(w=2) = \{N_1=3, N_2=2\}$.

STEP 3. k = 2일 때는 표 1과 같다.

여기서 $F_2(Q_2) > 0$ 이므로 최적해가 된다. 따라서 STEP 3를 $\Gamma(w=3)$에 대하여 반복할 필요 없이

RMAT의 최대값 = $F_2(Q_2) + (m - \overline{m}) \cdot MAT$
= 4 + (3-2) · 12
= 16

이 된다. 또한, 최적해는

표 1. K=2 일때의 결과

Q_2	R_2	$G^*(R_2)$	$F_1(Q_2\text{-}R_2)$	$\|U_1\|$	$\|U_2\|$	$f_3(E_2(R_2))$	$F_2(Q_2)$
{1, 2, 3, 4, 5}	{1, 2}	4	3	0	0	4	4
	{1, 3}	4	0	0	0	4	
	{1, 4}	5	M	0	0	-M	
	{1, 5}	5	M	0	0	-M	
	{2, 3}	M	1	0	0	-M	
	{2, 4}	3	1	0	0	3	
	{2, 5}	3	2	0	0	3	
	{3, 4}	5	M	0	0	-M	
	{3, 5}	4	1	0	0	4	
	{4, 5}	5	M	0	0	-M	

$R_1^* = \{1, 2, 4\}$, $R_2^* = \{3, 5\}$;
$R_1^* = \{2, 4, 5\}$, $R_2^* = \{1, 3\}$;
$R_1^* = \{3, 4, 5\}$, $R_2^* = \{1, 2\}$

이다. 최적 스케쥴은 STEP 1로부터 $R_1^* = \{1, 2, 4\}$, $R_2^* = \{3, 5\}$일 경우,

기계 1: $1 \rightarrow 2 \rightarrow 4$,
기계 2: $5 \rightarrow 3$

의 순서가 된다.

4. 계산 결과

본 연구에서 개발된 IDPMS의 문제해결 특성을 파악하고 DPMS와 비교 분석하기 위하여 다음과 같이 주어지는 자료를 이용하여 가상의 문제를 생성하였다:

준비시간 $S_{ij} \sim U(0, 100)$;
가공시간 $P_i \sim N(100, 30^2)$

여기서 U(a,b)는 하한이 a이고 상한이 b인 균등분포이고 $N(a,b^2)$는 평균이 a이고 표준편차가 b인 정규분포를 의미한다. 알고리즘의 문제해결 특성에 영향을 미치는 인자는 MAT와 작업수 n으로서

MAT = 330, 480, 530
n = 8, 10, 12, 14

의 각 경우를 고려하였다. 이러한 MAT와 N의 각 조합에 대하여 가상의 문제를 25개씩 임의로 생성한 후 각 문제에 대하여 IDPMS를 적용하여 풀어 본 결과가 표2에 정리되어 있다. 이 표로부터 MAT의 값이 증가함에 따라 1대의 기계에 최대로 할당 가능한 작업 수인 $\overline{N}$값이 증가하는 당연한 사실을 확인할 수 있다. 또한 n값이 증가함에 따라 평균 최대 소요 기억용량과 처리시간이 기하 급수적으로 증가함을 나타내고 있다. 따라서 소요 용량과 처리시간 측면에서 현장에서의 활용성을 평가하면 대략 14개 내외의 작업들을 최적으로 스케쥴링하는 데는 무난할 것으로 판단된다.

다음으로, IDMPS의 주요 특성 지표인 소요 기억 용량과 처리시간을 DPMS의 경우와 비교하여 보기 위하여 표2의 MAT=480일 경우에 DPMS를 적용하여 풀어 보았다(단, n=14일 경우에는 DPMS의 소요용량이 과다하여 적용할 수 없었음). 이 결과 표3에서 보는 바와 같이 소요용량과 처리시간 면에서 모두 IDPMS가 훨씬 우월한 것

으로 나타났다. 특히 소요 기억용량 면에서 보면 문제의 크기가 커짐에 따라 DPMS가 점점 많은 기억용량을 요구하고 있고 처리시간은 n>12이면 초과 비율이 엄청나게 증가 할 것으로 예상되어 IDPMS의 효용성이 입증되고 있다.

5. 결 론

본 연구에서는 순서의존적인 작업시간을 갖는 작업들을 병렬의 동일기계에서 처리할 때 잔여기계가용시간(RMAT)을 최대화하도록 각 기계에 할당된 작업들을 스케쥴링하는 문제를 다루었다. 이를 위하여, 수리적 모형을 수립하고 적은 규모에 대한 최적해를 생성할 수 있게 하기 위하여 동적계획법을 기초로 하여 대상 문제의 특성에 맞게 수정된 동적계획모형을 제시하였다.

본 연구에서 제시한 문제는 현장의 실정을 적극 반영한 것으로서 기존문헌에서 다루어진 바 없는 새로운 목적함수를 고려하였다는 데 의의가 있으며, 이에 대한 최초의 학문적 연구성과로서 가치가 있을 것으로 기대된다. 이를 통하여 산업현장에서는 주문량의 변화에 효과적으로 대처하는 작업 스케쥴링이 가능하게 되어 주어진 계획기간내에 수행할 작업들을 최소의 기계로 처리하고, 아울러 여유의 기계와 인적 자원을 다른 생산 작업에 활용할 수 있을 것이다.

참고문헌

[1] 강 상구, "다품종 소량 생산하의 작업순서 결정기준에 관한 연구," 울산대학교 산업경영 대학원 산업관리공학과 석사학위 논문, 1991.

표 2. IDPMS에 의한 문제 적용 결과

	MAT											
	330				480				530			
n	8	10	12	14	8	10	12	14	8	10	12	14
평균 $\overline{N}$	2.9	3.4	3.2	3.0	4.3	4.6	4.6	4.7	4.6	5.0	5.3	5.2
평균 최대 소요 기억 용량(byte)	1,454.9	4,384.0	13,330.6	33,964.0	2,132.6	7,948.6	25,884.2	75,604.9	2,272.9	9,111.1	33,252.6	96,634.9
평균 처리 시간 (CPU sec.)	0.27	2.76	17.18	160.97	0.29	2.73	17.53	199.80	0.20	3.12	17.64	254.94

표 3. IDPMS와 DPMS의 비교

	n								
	8			10			12		
	DPMS	IDPMS	초과 비율*	DPMS	IDPMS	초과 비율*	DPMS	IDPMS	초과 비율*
평균 최대 메모리 소요량(byte)	7,644.0	2,132.6	258.4	32,728.0	7,948.6	311.7	126,936.0	25,884.2	390.4
평균 처리 시간 (CPU sec.)	0.38	0.29	31.0	3.45	2.73	26.4	31.98	17.53	82.4

* 초과 비율(%) = (DPMS - IDPMS) · 100 / IDPMS

[2] 임진규, 박주철, "절단압착공정의 기계별 작업순서 결정방법개선에 관한 연구," 산업공학, 제9권 2호, pp. 83-94, 1996.

[3] Cheng, T. C. E., Chen, Z. L., Kovalyav, M. Y. and Lin, B. M. T., "Parallel - machine Batching and Scheduling to Minimize Total Completion Time," *IIE Transactions*, Vol. 28, pp. 953-956, 1996.

[4] Cheng, T. C. E. and Sin, C. C. S., "A State-of-the-Art Review of Parallel-Machine Scheduling Research," *European Journal of Operational Research*, Vol. 47, pp. 271-292, 1990.

[5] Dearing, P. M. and Henderson, R. A., "Assigning Looms in a Textile Weaving Operation with Change-over Limitation," *Prod. Inv. Mgmt.*, Vol. 25, pp. 23-31, 1984

[6] Dogramaci, A., "Production Scheduling of Independent Jobs on Parallel Identical Processors," *International Journal of Production Research*, Vol. 22, No. 4, pp 535-548, 1984.

[7] Franca, M, P., Gendreau, M., Laporte, G., and Müller, F. M., "A Tabu Search Heuristic for the Multiprocessor Scheduling Problem with Sequence Dependent Setup Times," *International Journal of Production Economics*, 43, pp. 79-89, 1996.

[8] Frederickson, G., Hecht, M. S., and Kim, C. E., "Approximation Algorithm for Some Routing Problems," *SIAM*, Vol. 7, pp. 178-193, 1978.

[9] Gavish, B. and Srikanth, K., "An Optimal Solution Method for Large-scale Multiple Traveling Salesmen Problems," *Operations Research*, Vol. 34, No. 5, pp. 698-717, 1986.

[10] Guinet, A., "Textile Production Systems: A Succession of Non-identical Parallel Processor Shops," *J. of Opl. Res. Soc.*, Vol. 42, pp. 655-671, 1991.

[11] Held, M. and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society of Industrial and Applied Mathmatics*, Vol. 10, pp. 196-210, 1962.

[12] Ovacik, I. M. and Uzsoy, R., "Worst-Case Error Bounds for Parallel Machine Scheduling Problems with Bounded Sequence-Dependent Setup Times," *Operations Research Letters*, Vol. 14, pp. 251-256, 1993.

[13] Parker, R. G., Deane, R. H., and Holmes, R. A., "On the Use of a Vehicle Routing Algorithm for the Parallel Processor Problems with Sequence Dependent Changeover Costs," *AIIE Trans*, Vol. 9, No. 12, pp. 155-160, 1977.

[14] Pattloch, M. and Schmidt, G., "Lot-size Scheduling of Two Types of Jobs on Identical Machines," *Discrete Applied Mathematics*, Vol. 65, pp. 409-419, 1996.

[15] Serafini, P. and Speranza, M. G., "Production Scheduling Problems in a Textile Industry," *European Journal of Operations Research*, 58, pp. 173-190, 1992.

[16] Sumichrast, R. and Baker, J. R., "An Integer Linear Programming based Heuristic for Minimizing Setup Time," *International Journal of Production Research*, Vol. 25, No.5, pp. 761-771, 1987.

98년 5월 최초 접수, 98년 6월 최종 수정

〈부록〉

아래와 같이 자료가 주어지는 문제에 대하여 DPMS를 적용하여 보자.

$n = 5,\ m = 3,\ MAT = 12,$

$$S_{ij} = \begin{bmatrix} - & 1 & 2 & 3 & 2 \\ 2 & - & 4 & 2 & 5 \\ 1 & 5 & - & 2 & 1 \\ 2 & 4 & 1 & - & 4 \\ 3 & 2 & 2 & 3 & - \end{bmatrix}$$

$SM_i = (1\ 2\ 3\ 2\ 1),$
$P_i = (2\ 4\ 3\ 1\ 2).$

이 문제의 풀이 과정을 단계별로 도표로 정리하면 다음과 같다:

1) K = 1 일 때

R	J_q	t_{iq}	$G_q(R)$	$G^*(R)$
{1}	1	1 + 2	9	9
{2}	2	2 + 4	6	6
{3}	3	3 + 3	6	6
{4}	4	2 + 1	9	9
{5}	5	1 + 2	9	9

R	q	t_{iq}	$G_i(R\text{-}\{J_q\})$	$G_q(R)$	$G^*(R)$
{1, 2}	1	2 + 2	6	2	4
	2	1 + 4	9	4	
{1, 3}	1	1 + 2	6	3	4
	3	2 + 3	9	4	
{1, 4}	1	2 + 2	9	5	5
	4	3 + 1	9	5	
{1, 5}	1	3 + 2	9	4	5
	5	2 + 2	9	5	
{2, 3}	2	5 + 4	6	-M	M
	3	4 + 3	6	-M	
{2, 4}	2	4 + 4	9	1	3
	4	2 + 1	6	3	
{2, 5}	2	2 + 4	9	3	3
	5	5 + 2	6	-M	
{3, 4}	3	1 + 3	9	5	5
	4	2 + 1	6	3	
{3, 5}	3	2 + 3	9	4	4
	5	1 + 2	6	3	
{4, 5}	4	3 + 1	9	5	5
	5	4 + 2	9	3	

R	q	t_q	$G_i(R-\{J_q\})$	$G_q(R)$	$G^*(R)$
{1, 2, 3}	1	2 + 2	M	-M	M
		1 + 2	M	-M	
	2	1 + 4	3	-M	
		5 + 4	4	-M	
	3	2 + 3	2	-M	
		4 + 3	4	-M	
{1, 2, 4}	1	2 + 2	1	-M	1
		2 + 2	3	-M	
	2	2 + 4	5	-M	
		4 + 4	5	-M	
	4	3 + 1	2	-M	
		2 + 1	4	1	
{1, 2, 5}	1	2 + 2	3	-M	M
		3 + 2	M	-M	
	2	1 + 4	4	-M	
		2 + 4	5	-M	
	5	2 + 2	2	-M	
		5 + 2	4	-M	
{1, 3, 4}	1	1 + 2	5	2	2
		2 + 2	3	-M	
	3	2 + 3	5	0	
		1 + 3	5	1	
	4	3 + 1	3	-M	
		2 + 1	4	1	
{1, 3, 5}	1	1 + 2	4	1	1
		3 + 2	3	-M	
	3	2 + 3	4	0	
		2 + 3	5	0	
	5	2 + 2	3	-M	
		1 + 2	4	1	
{1, 4, 5}	1	2 + 2	5	1	1
		3 + 2	3	-M	
	4	3 + 1	4	0	
		3 + 1	5	1	
	5	2 + 2	5	1	
		4 + 2	5	-M	
{2, 3, 4}	2	5 + 4	5	-M	M
		4 + 4	3	-M	
	3	4 + 3	1	-M	
		4 + 3	3	-M	
	4	2 + 1	M	-M	
		2 + 1	M	-M	
{2, 3, 5}	2	5 + 4	4	-M	M
		2 + 4	3	-M	
	3	4 + 3	3	-M	
		2 + 3	M	-M	
	5	5 + 1	M	-M	
		1 + 1	M	-M	
{2, 4, 5}	2	4 + 4	5	-M	0
		2 + 4	3	-M	
	4	2 + 1	3	0	
		3 + 1	M	-M	
	5	5 + 1	1	-M	
		4 + 1	3	-M	
{3, 4, 5}	3	1 + 3	5	1	3
		2 + 3	3	-M	
	4	2 + 1	4	-M	
		3 + 1	3	-M	
	5	1 + 1	5	3	
		4 + 1	3	-M	

$|R| = 4$인 모든 R에 대하여 $G^*(R) = M$ 이다.

2) k = 2 일 때

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\|U_1\|$	$\|U_2\|$	E_2	$F_2(Q_2)$
{1}	ϕ	12	9	-	-	21	21
	{1}	9	12	-	-	21	
{2}	ϕ	12	6	-	-	18	18
	{2}	6	12	-	-	18	
{3}	ϕ	12	6	-	-	18	18
	{3}	6	12	-	-	18	
{4}	ϕ	12	9	-	-	21	21
	{4}	9	12	-	-	21	
{5}	ϕ	12	9	-	-	21	21
	{5}	9	12	-	-	21	
{1, 2}	ϕ	12	4	-	-	16	16
	{1}	9	6	0	0	9	
	{2}	6	9	0	0	9	
	{1, 2}	4	12	-	-	16	
{1, 3}	ϕ	12	4	-	-	16	16
	{1}	9	6	0	0	9	
	{3}	6	9	0	0	9	
	{1, 3}	4	12	-	-	16	
{1, 4}	ϕ	12	5	-	-	17	17
	{1}	9	9	0	0	9	
	{4}	9	9	0	0	9	
	{1, 4}	5	12	-	-	17	
{1, 5}	ϕ	12	5	-	-	17	17
	{1}	9	9	0	0	9	
	{5}	9	9	0	0	9	
	{1, 5}	5	12	-	-	17	

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\|U_1\|$	$\|U_2\|$	E_2	$F_2(Q_2)$
{2, 3}	ϕ	12	M	-	-	M	6
	{2}	6	6	0	0	6	
	{3}	6	6	0	0	6	
	{2, 3}	M	12	-	-	M	
{2, 4}	ϕ	12	3	-	-	15	15
	{2}	6	9	0	0	9	
	{4}	9	6	0	0	9	
	{2, 4}	3	12	-	-	15	
{2, 5}	ϕ	12	3	-	-	15	15
	{2}	6	9	0	0	9	
	{5}	9	6	0	0	9	
	{2, 5}	3	12	-	-	15	
{3, 4}	ϕ	12	5	-	-	17	17
	{3}	6	9	0	0	9	
	{4}	9	6	0	0	9	
	{3, 4}	5	12	-	-	17	
{3, 5}	ϕ	12	4	-	-	16	16
	{3}	6	9	0	0	9	
	{5}	9	6	0	0	9	
	{3, 5}	4	12	-	-	16	
{4, 5}	ϕ	12	5	-	-	17	17
	{4}	9	9	0	0	9	
	{5}	9	9	0	0	9	
	{4, 5}	5	12	-	-	17	
{1, 2, 3}	ϕ	12	M	-	-	M	9
	{1}	9	M	0	0	M	
	{2}	6	4	0	0	6	
	{3}	9	4	0	0	9	
	{1, 2}	4	9	0	0	9	
	{1, 3}	4	6	0	0	6	
	{2, 3}	M	9	0	0	M	
	{1, 2, 3}	M	12	-	-	M	

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\lvert U_1\rvert$	$\lvert U_2\rvert$	E_2	$F_2(Q_2)$
{1, 2, 4}	ϕ	12	1	-	-	13	13
	{1}	9	3	0	0	9	
	{2}	6	5	0	0	6	
	{4}	9	4	0	0	9	
	{1, 2}	4	9	0	0	9	
	{1, 4}	5	6	0	0	6	
	{2, 4}	3	9	0	0	9	
	{1, 2, 4}	1	12	-	-	13	
{1, 2, 5}	ϕ	12	M	-	-	M	9
	{1}	9	3	0	0	9	
	{2}	6	5	0	0	6	
	{5}	9	4	0	0	9	
	{1, 2}	4	9	0	0	9	
	{1, 5}	5	6	0	0	6	
	{2, 5}	3	9	0	0	9	
	{1, 2, 5}	M	12	-	-	M	
{1, 3, 4}	ϕ	12	2	-	-	14	14
	{1}	9	5	0	0	9	
	{3}	6	5	0	0	6	
	{4}	9	4	0	0	9	
	{1, 3}	4	9	0	0	9	
	{1, 4}	5	6	0	0	6	
	{3, 4}	5	9	0	0	9	
	{1, 3, 4}	2	12	-	-	14	
{1, 3, 5}	ϕ	12	1	-	-	13	13
	{1}	9	4	0	0	9	
	{3}	6	5	0	0	6	
	{5}	9	4	0	0	9	
	{1, 3}	4	9	0	0	9	
	{1, 5}	5	6	0	0	6	
	{3, 5}	4	9	0	0	9	
	{1, 3, 5}	1	12	-	-	13	

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\lvert U_1\rvert$	$\lvert U_2\rvert$	E_2	$F_2(Q_2)$
{1, 4, 5}	ϕ	12	1	-	-	13	13
	{1}	9	5	0	0	9	
	{4}	9	5	0	0	9	
	{5}	9	5	0	0	9	
	{1, 4}	5	9	0	0	9	
	{1, 5}	5	9	0	0	9	
	{4, 5}	5	9	0	0	9	
	{1, 4, 5}	1	12	-	-	13	
{2, 3, 4}	ϕ	12	M	-	-	M	6
	{2}	6	5	0	0	6	
	{3}	6	3	0	0	6	
	{4}	9	M	0	0	M	
	{2, 3}	M	9	0	0	M	
	{2, 4}	3	6	0	0	6	
	{3, 4}	5	6	0	0	6	
	{2, 3, 4}	M	12	-	-	M	
{2, 3, 5}	ϕ	12	M	-	-	M	6
	{2}	6	4	0	0	6	
	{3}	6	3	0	0	6	
	{5}	9	M	0	0	M	
	{2, 3}	M	9	0	0	M	
	{2, 4}	3	6	0	0	6	
	{3, 5}	4	6	0	0	6	
	{2, 3, 5}	M	12	-	-	M	
{2, 4, 5}	ϕ	12	0	-	-	12	12
	{2}	6	5	0	0	6	
	{4}	9	3	0	0	9	
	{5}	9	3	0	0	9	
	{2, 4}	3	9	0	0	9	
	{2, 5}	3	9	0	0	9	
	{4, 5}	5	6	0	0	6	
	{2, 4, 5}	0	12	-	-	12	

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\lvert U_1 \rvert$	$\lvert U_2 \rvert$	E_2	$F_2(Q_2)$
{3, 4, 5}	ϕ	12	3	-	-	15	15
	{3}	6	5	0	0	6	
	{4}	9	4	0	0	9	
	{5}	9	5	0	0	9	
	{3, 4}	5	9	0	0	9	
	{3, 5}	4	9	0	0	9	
	{4, 5}	5	6	0	0	6	
	{3, 4, 5}	3	12	-	-	15	
{1, 2, 3, 4}	ϕ	12	M	-	-	M	6
	{1}	9	M	0	0	M	
	{2}	6	2	0	0	6	
	{3}	6	1	0	0	6	
	{4}	9	M	0	0	M	
	{1, 2}	4	5	0	0	5	
	{1, 3}	4	3	0	0	4	
	{1, 4}	5	M	0	0	M	
	{2, 3}	M	5	0	0	M	
	{2, 4}	3	4	0	0	4	
	{3, 4}	5	4	0	0	5	
	{1, 2, 3}	M	9	0	0	M	
	{1, 2, 4}	1	6	0	0	6	
	{1, 3, 4}	2	6	0	0	6	
	{2, 3, 4}	M	9	0	0	M	
	{1, 2, 3, 4}	M	12	-	-	M	
.							

Q_2	R_2	$G^*(R_2)$	$F_1(Q_1)$	$\lvert U_1 \rvert$	$\lvert U_2 \rvert$	E_2	$F_2(Q_2)$
{1, 2, 3, 4, 5}	ϕ	12	M	-	-	M	4
	{1}	9	M	0	0	M	
	{2}	6	M	0	0	M	
	{3}	6	M	0	0	M	
	{4}	9	M	0	0	M	
	{5}	9	M	0	0	M	
	{1, 2}	4	3	0	0	4	
	{1, 3}	4	0	0	0	4	
	{1, 4}	5	M	0	0	M	
	{1, 5}	5	M	0	0	M	
	{2, 3}	M	1	0	0	M	
	{2, 4}	3	1	0	0	3	
	{2, 5}	3	2	0	0	3	
	{3, 4}	5	M	0	0	M	
	{3, 5}	4	1	0	0	4	
	{4, 5}	5	M	0	0	M	
	{1, 2, 3}	M	5	0	0	M	
	{1, 2, 4}	1	4	0	0	4	
	{1, 2, 5}	M	5	0	0	M	
	{1, 3, 4}	2	3	0	0	3	
	{1, 3, 5}	1	3	0	0	3	
	{1, 4, 5}	1	M	0	0	M	
	{2, 3, 4}	M	5	0	0	M	
	{2, 3, 5}	M	5	0	0	M	
	{2, 4, 5}	0	4	0	0	4	
	{3, 4, 5}	3	4	0	0	4	
	{1, 2, 3, 4}	M	9	0	0	M	
	{1, 2, 3, 5}	M	9	0	0	M	
	{1, 2, 4, 5}	M	6	0	0	M	
	{1, 3, 4, 5}	M	6	0	0	M	
	{2, 3, 4, 5}	M	9	0	0	M	
	{1, 2, 3, 4, 5}	M	12	-	-	M	

3) k = 3 일 때

Q_3	R_3	$G^*(R_3)$	$F_2(Q_2)$	$\|U_2\|$	$\|U_3\|$	E_2	$F_3(Q_3)$
{1, 2, 3, 4, 5}	ϕ	12	4	-	-	16	16
	{1}	9	6	0	0	9	
	{2}	6	9	0	0	9	
	{3}	6	9	0	0	9	
	{4}	9	6	0	0	9	
	{5}	9	6	0	0	9	
	{1, 2}	4	15	1	1	16	
	{1, 3}	4	12	1	1	16	
	{1, 4}	5	9	0	0	9	
	{1, 5}	5	6	0	0	6	
	{2, 3}	M	13	1	1	M	
	{2, 4}	3	13	1	1	15	
	{2, 5}	3	14	1	1	15	
	{3, 4}	5	9	0	0	9	
	{3, 5}	4	13	1	1	16	
	{4, 5}	5	9	0	0	9	
	{1, 2, 3}	M	17	0	0	M	
	{1, 2, 4}	1	16	1	1	16	
	{1, 2, 5}	M	17	1	1	M	
	{1, 3, 4}	2	15	1	1	15	
	{1, 3, 5}	1	15	1	1	15	
	{1, 4, 5}	1	6	0	0	6	
	{2, 3, 4}	M	17	1	1	M	
	{2, 3, 5}	M	17	1	1	M	
	{2, 4, 5}	0	16	1	1	16	
	{3, 4, 5}	3	16	1	1	16	
	{1, 2, 3, 4}	M	21	1	1	M	
	{1, 2, 3, 5}	M	21	1	1	M	
	{1, 2, 4, 5}	M	18	1	1	M	
	{1, 3, 4, 5}	M	18	1	1	M	
	{2, 3, 4, 5}	M	21	1	1	M	
	{1, 2, 3, 4, 5}	M	24	-	-	M	

여기서 RMAT의 최대값은 $F_3(Q_3=J)$ = 16이 되며, 최적해는 다음과 같다:

R_1^* = {1, 2, 4}, R_2^* = {3, 5}, $R_3^* = \phi$; R_1^* = {3, 5}, R_2^* = {1, 2, 4}, $R_3^* = \phi$;
R_1^* = {1, 2, 4}, $R_2^* = \phi$, R_3^* = {3, 5}; R_1^* = {3, 5}, $R_2^* = \phi$, R_3^* = {1, 2, 4};
$R_1^* = \phi$, R_2^* = {1, 2, 4}, R_3^* = {3, 5}; $R_1^* = \phi$, R_2^* = {3, 5}, R_3^* = {1, 2, 4};
R_1^* = {2, 4, 5}, R_2^* = {1, 3}, $R_3^* = \phi$; R_1^* = {1, 3}, R_2^* = {2, 4, 5}, $R_3^* = \phi$;
R_1^* = {2, 4, 5}, $R_2^* = \phi$, R_3^* = {1, 3}; R_1^* = {1, 3}, $R_2^* = \phi$, R_3^* = {2, 4, 5};
$R_1^* = \phi$, R_2^* = {2, 4, 5}, R_3^* = {1, 3}; $R_1^* = \phi$, R_2^* = {1, 3}, R_3^* = {2, 4, 5};
R_1^* = {3, 4, 5}, R_2^* = {1, 2}, $R_3^* = \phi$; R_1^* = {1, 2}, R_2^* = {3, 4, 5}, $R_3^* = \phi$;
R_1^* = {3, 4, 5}, $R_2^* = \phi$, R_3^* = {1, 2}; R_1^* = {1, 2}, $R_2^* = \phi$, R_3^* = {3, 4, 5};
$R_1^* = \phi$, R_2^* = {3, 4, 5}, R_3^* = {1, 2}; $R_1^* = \phi$, R_2^* = {1, 2}, R_3^* = {3, 4, 5};

이러한 복수해 중에서, R_1^* = {1, 2, 4}, R_2^* = {3, 5}, $R_3^* = \phi$일 경우, k = 1인 경우로부터 그 최적 순서를 알 수 있는데

기계 1 : 1 → 2 → 4;
기계 2 : 5 → 3

의 순이 된다.