

자원비용을 고려한 프로젝트 스케줄링에 관한 연구

-Project Scheduling Problem with Resource Constraints Minimizing Cost-

서 순 근*
최 종 덕**

Abstract

In this paper, constrained resource project scheduling problems schedule project activities subject to finite constraints on the availability of non-storable resource.

Further, resources are assumed to be available per period in constant amounts, and are also demanded by an activity in constant amounts throughout the duration of the activity. We describe formulation which minimizes the combined cost of fluctuations in resource demand and delay of project completion. Cost bounding procedures are augmented by dominance relationships presented as theorems.

This paper presents algorithm for solving the problem. And numerical examples are presented. Sensitivity analysis to evaluate the effect of changes of cost coefficients is conducted.

1. 서 론

경제 성장과 더불어 국내 기업들의 해외진출은 급속히 증가하고 국내의 임금비 상승과 근로 조건의 개선으로 산업인력의 수급은 더욱 압박을 받고 있는 실정이다.

이러한 환경하에서 수행되는 프로젝트를 성공적으로 완수하기 위하여 자원을 효율적으로 사용할 필요가 있다. 현실적으로 자원이 충분한 경우는 극히 드물고 대부분의 경우 자원에 제약이 있는 프로젝트 계획 및 통제를 하게 된다. 자원을 합리적이고 과학적으로 관리한다는 것은 기업경영에서는 원가와 생산성에 직결되는 문제이므로 매우 중요하다.

프로젝트 스케줄링문제(project scheduling problem)은 주어진 선행관계(precedence relation)에 의해서 행하여지는 작업군(activities)에서 어떤 잡(job) 혹은 계획에 대해 자원[기계, 장비, 설비, 인력 외에도 공학적 기술, 경영기술 및 자본 등]의 제약이 없는 경우 또는 제약이 있는 경우에 있어서 다음과 같은 두 가지 형으로 분류된다. 그 중 하나는 프로젝트시간(총 소요시간)과 같은 목적함수를 최소로 하는 스케줄을 구하는 PERT(Program Evaluation and Review Technique)형 문제이고 자원평활문제(resource levelling(balancing) problem), 자원배분문제(resource allocation problem)은 이 형에 속한다. 또 하나는 각 작업에서 요하는 비용을 주었을 때에 지정시간내에서 최소 비용의 스케줄을 구하는 CPM(Critical Path Method)형 문제이다.

이 기법은 1950년대에 미국에서 개발되어 현재 그 영역의 확대와 응용면이 더욱 활발해져 우주개발계획, 원자로의 건설계획, 신규공장의 건설계획, 도로공사 및 댐 건설 등에서 널리 이용되고 있다. 오늘날 프로젝트를 실행할 경우에 복잡한 선행관계가 존재하며 실행시에 이용되는 자원도 다양화되고 있다. 지금까지 국내에서 발표된 연구들은 대부분이 서술적이며 실증적인 연구는 별로 없는 실정이고 프로젝트 관리 시스템의 부재로 많은 자원의 낭비를 초래하고 있으면서도 아직 이에 대응하는 국내의 연구는 미약하다. 이 문제는 많은 계산시간을 요하므로 발전적 기법[1, 2, 7, 10]이 많이 연구되고 자원제약형태도 다수[1, 2, 5, 6, 8]이나 주로 자원제약 조건을 만족하면서 낭비를 최소화하는 모형이 대부분이며 제약조건과 낭비를 동시에 고려한 연구는 드문 실정이다.

그러나 현실적으로 공정을 조정해서 수준값을 가능한 한 낮게해서 비용을 최소화하는 것은 기업에 있어서 중요하다.

*동아대학교 공과대학 산업공학과 교수

**동아대학교 대학원 산업공학과

접수: 1991. 4. 27.

따라서 본 연구에서는 먼저 항상 사용 가능한 자원에 대해 수준(level)을 설정하고 그 자원 수준의 제약에 대해 제약조건을 초과하는 비용(잔업비용, 특근 및 하청 등), 제약 수준미달시에 유희자원을 최소화하는 적정 수준을 유지하기 위한 유희비용인 벌금(penalty)과 납기가 정해져 있어 납기 지연에 따른 지연비용의 합을 최소화 하는 모형을 설정한 후 이의 해법의 효율성을 검토한다. 또한 제시된 해법의 실용성을 보이기 위해 수치 실험에 의한 결과를 고찰한다.

2. 가정과 기호 설명

2.1 본 연구의 가정

종래의 연구에서는 항상 사용 가능한 자원에 대한 비용, 그 외의 자원비용(잔업비용)을 고려한 모형은 드물고, 자원제약을 초과하지 않는 스케줄링 모형이 많다.

또 자원의 peak치를 최소로 하는 연구도 있지만 peak치만 고려하고 그 외의 자원 비용에 대해서는 고려하고 있지 않다. 따라서 본 연구에서는 peak치보다는 항상 사용 가능한 자원 조건하에서 프로젝트 기간을 통해서 자원에 관련된 비용을 포함하면서 납기지연을 최소화 하는 스케줄링 모형을 설정한다.

본 연구에서 가정은 다음과 같다.

- 1) 프로젝트 스케줄을 행한다.
- 2) 프로젝트는
 - a) 다수의 작업(Activity)으로 구성된다.
 - b) 납기가 정해져 있다. (납기에 지연되는 경우에는 벌금(penalty)를 부과한다.
- 3) 프로젝트 중에서 각각의 작업은
 - a) 기지소요시간을 갖는다.
 - b) 선행하는 작업이 완료할 때까지 개시할 수 없다.
 - c) 특정 종류의 자원을 미리 정해진 수준을 필요로 한다.
 - d) 예외적인 경우를 제외하고는 중단할 수 없다.
- 4) 프로젝트 기간을 통해 항상 사용할 수 있는 자원량은 정해져 있다.
 - a) 그 자원량을 초과하는 경우에 벌금(잔업에 의한 비용)을 부과한다.
 - b) 그 자원량에 미달하는 경우에 벌금(유희에 의한 비용)을 부과한다.

이러한 조건하에서 본 연구에서는 납기지연에 의한 비용과 항상 사용 가능한 자원 비용에 수준을 초과하는 비용(잔업비용)과 미달시 유희비용을 합친 총 비용을 최소화 한다.

2.2 기호설명

- L : 수준(level)
- W : 납기 지연에 의한 비용(벌금)
- A : L을 초과하는 비용
- B : L에 미달시의 비용
- C : 종료시각
- d : 납기
- V : 자원의 1기간 1단위에 드는 비용
- r_{jt} : 기간 t에서 job의 자원 사용량
- P_j : Job j의 처리 시간
- t_j : Job j의 개시 시간
- $J(t)$: 기간 t에서 처리되는 Job의 집합

3. 본 연구의 해법

3.1 본 연구의 모형화

일반적인 비용은 기계, 설비비, 재료비, 노무비 등으로 이루어지지만 각 작업의 소요재료량, 작업량 자체는

개개의 작업 특수성에 의존하는 것이므로 공정 최적화 단계에서 주어진다.

따라서 본 연구에서는 먼저 최초에 항상 가능한 자원량의 수준 L 을 설정하고 다음과 같은 목적함수를 Job의 선행관계를 지키면서 최소가 되는 것을 생각한다.

$$(납기지연비용) + (항상 사용 가능한 자원에 대한 비용) + (수준에 초과하는 자원에 대한 비용) + (수준에 미달하는 자원에 비용) \tag{1}$$

이 (1)식을 다음과 같이 정식화할 수 있다.

$$W \{ \max(0, c-d) \} + VcL + A \{ \max(0, \sum r_{jt} - L) \} + B \{ \max(0, L - \sum r_{jt}) \} \tag{2}$$

3.2 특성과 해법

먼저 정식화 할 시에는 최초에 열거해야 하는 프로젝트 스케줄(job순)의 정의에 대해 설명한다.

[정의 1]

프로젝트를 구성하고 있는 모든 Job의 선행관계를 만족하는 스케줄을 프로젝트 스케줄(job순)이라 한다.

n 개의 Job의 가능한 프로젝트 스케줄(job순)을 열거하고 그것을 모든 프로젝트 스케줄(job순)에 대해서 다음과 같은 순서로 계산을 행한다.

이 프로젝트 스케줄(job순)에서 j 번째에 처리되는 Job을 Job $j(j=1, \dots, n)$ 라 한다. (2)식의 제 1항과 제 2항은 종료시각 c 에 의한 함수이므로 이 두 항을 합해 $g(c)$ 로 표시한다. 자원은 k 종류이고, 자원 k 의 수준 L_k ($k=1, \dots, k$)라 하고 L 을 초과하는 벌금을 A_k , 미달시에 벌금을 B_k 라 하며 Job j 의 단위기간에서 자원 k 의 사용량을 r_{jk} 로 하면 (2)식은 다음과 같이 된다.

$$g(C) + \sum_t \sum_k [A_k \{ \max(0, \sum J(t)r_{jk} - L_k) \} + B_k \{ \max(0, L_k - \sum J(t)r_{jt}) \}]$$

여기서 $J(t)$ 를 기간 t 에서 처리되는 Job의 집합이다. 또 함수 $g(C)$ 는 다음과 같은 함수이다.

$$g(C) = W \{ \max(0, c-d) \} + \sum_k \cdot V_k \cdot c \cdot L_k$$

Job j 의 처리시간을 P_j , 개시시간을 t_j 라 하며 또 α_j 를 Job j 의 개시 시점에서 네트워크 종점까지의 최장 path의 길이로 한다.

[정의 2]

t_1, t_2, \dots, t_m 가 $t_i \geq t_s (s \in p(j))$ 와 $t_i + \alpha_j \leq T (j=1, 2, \dots, m)$, (T 는 종료시각 상계치)을 만족하는 비수 정수이면, m 벡터 $S_1 = (t_1, t_2, \dots, t_m)$ 은 m 부분 스케줄이다. 여기서 $P(j)$ 는 Job j 에 선행하는 Job의 집합이다.

[정의 3]

(S_m, F_m) 가 프로젝트 스케줄이면 $n-m$ 벡터 $F_m = (t_{m+1}, t_{m+2}, \dots, t_n)$ 은 S_m 의 $n-m$ 의 부분스케줄이다. 알고리즘에서는 최소의 하계치로 프로젝트 스케줄이 종료 가능하가를 검토하면서 부분 스케줄을 조사해 간다. 그 과정에서 다음의 함수를 이용한다.

함수 β 는 Job j 를 실행할 시 자원부하의 합계로써

$$\beta(t_j) = \sum_t \sum_k [A_k \{ \max(0, \sum J(t)r_{jk} - L_k) \} + B_k \{ \max(0, L_k - \sum J(t)r_{jt}) \}]$$

이고, $\beta(S_m) = \beta(t_1) + \beta(t_2) + \dots + \beta(t_m)$ 이다. m 부분 스케줄의 하계치의 함수는

$$LB(S_m) = \beta(S_m) + g[\max_{j=1, 2, \dots, m} (t_j + \alpha_j)]$$

$LB(S_n)$ 은 프로젝트 스케줄 S_n 의 하계치이다.

$LB(S_m)$ 은 S_m 을 종료하는 것에 의해서 얻어지는 m 부분 스케줄의 하계치이다.

[정의 4]

모든 F_m 에 대해서 $LB(S_m, \bar{F}_m) \leq LB(S_m, F_m)$ 을 만족하면 $n-m$ 벡터 $\bar{F}_m = (t_{m+1}, t_{m+2}, \dots, t_n)$ 은 최적인

n-m부분 스케줄이다.

[정의 5]

프로젝트 스케줄 S_n 에 대해서 $LB(\bar{S}_n) < LB(S_n)$ 을 만족하면, n벡터 S_n 은 최적 프로젝트 스케줄이다.

[정의 6]

S_m 과 S'_m 을 두 개의 m부분 스케줄이라 하고, \bar{F}_m, \bar{F}'_m 을 그것들의 최적 n-m부분 스케줄이라 한다. 이때 $LB(S_m, \bar{F}_m) < LB(S'_m, \bar{F}'_m)$ 이면 S_m 은 S'_m 에 우월한 것이 명백하다.

최적 프로젝트 스케줄을 결정하기 위해 부분 스케줄을 조사할 시에 어떤 부분 스케줄이 다른 부분 스케줄에 우월하면 그 부분 스케줄을 무시할 수 있다.

우월성에 관해 다음과 같은 정리를 이용한다.

[정리 1]

S_m^1, S_m^2 와 $\beta(S_m^1) < \beta(S_m^2)$ 이면, S_m^1 은 S_m^2 에 우월한다.

증명 : \bar{F}_m^1, \bar{F}_m^2 을 m부분 스케줄 S_m^1, S_m^2 의 최적 n-m부분 스케줄이라 한다.

$S_m^1 < S_m^2$ 은 \bar{F}_m^2 가 S_m^1 에 대해서 실행가능에 있으므로 다음과 같이 된다.

$$\begin{aligned} LB(S_m^1, \bar{F}_m^1) &\leq LB(S_m^1, \bar{F}_m^2) \\ &= \beta(S_m^1) + \beta(\bar{F}_m^2) + g[\max\{\max_{j=1, 2, \dots, m}(t_j + \alpha_j), \max_{j=m+1, m+2, \dots, n}(t_j + \alpha_j)\}] \leq \beta(S_m^2) \\ &\quad + \beta(F_m^2) + g[\max_{j=1, 2, \dots, n}(t_j + \alpha_j)] \\ &= LB(S_m^2, F_m^2) \end{aligned}$$

다음은 최적 프로젝트 스케줄을 발견하는 알고리즘을 찾는다. β 와 LB의 값에 의해 부분 스케줄의 집합 M을 구축한다. 최초 M은 m=1인 부분 스케줄만을 포함하고 m=n에서 최적 스케줄이 발견되도록 m값을 크게해서 M에 부분 스케줄을 더해간다. M을 증가시키는 경우 M에서 다른 스케줄에 우월하고 있지 않는 부분 스케줄 S_m 만 더한다.

본 연구의 알고리즘은 다음과 같다.

(알고리즘)

step 1) RERT을 계산하고 그때의 자원량의 PEAK치를 L_{MAX} 로 한다.

step 2) L을 설정한다.

step 3) 설정했던 L에 관해서 (2)식을 최소화 한다.

step 4) $M = \emptyset$ 라 하고 열거한 프로젝트 스케줄의 JOB 1의 개시시간 t_1 을 0라 해서 $\bar{s}=(t_1)$ 을 M에 더한다.

step 5) 만일, $m=n-1$ 이면 step 7)로 간다.

M에서 S_m 을 제외하고 $t_{m+1}=t_m, \dots, \max_{j \in p(m+1)}(t_j + p_j)$ 에 의해서 주어지는 t_{m+1} 값의 집합을 고려한다.

$t'_{m+1} > t_{m+1}, \beta(t'_{m+1}) > \beta(t_{m+1})$ 로 되는 모든 t'_{m+1} 을 제외하고 그외의 t_{m+1} 에 대해서는 (S_m, t_{m+1}) 을 M에 더한다.

모든 (S_m, t_{m+1}) 에 대해서, $LB(\overline{S_m, t_{m+1}}) < LB(S_m, t_{m+1})$ 을 만족하도록 (S_m, t_{m+1}) 을 M에서 부분 스케줄 $\overline{S_{m+1}}$ 로 한다.

step 6) $m=m+1$ 로 하고 step 4)로 간다.

step 7) ($m=n-1$)

$t_n=t_{n-1}, \dots, \max_{j \in p(n)}(t_j + p_j)$ 에 의해서 주어지는 t_n 값의 모든 것에 대해서

$LB(\overline{S_{n-1}, t_n}) < LB(S_{n-1}, t_n)$ 로 되는 (S_{n-1}, t_n) 을 발견한다.

이 스케줄을 최적 스케줄이라 한다.

step 8) 설정했던 L을 $[0, L_{max}]$ 의 범위에서 변경하고 step 3)으로 간다.

4. 수치 예

간단한 단일 자원인 경우에 대한 수치예의 네트워크는 그림 1과 같다.
이것을 이용하여 각 기간에 대한 자원소요량을 그래프로 그리면 그림 2와 같다.

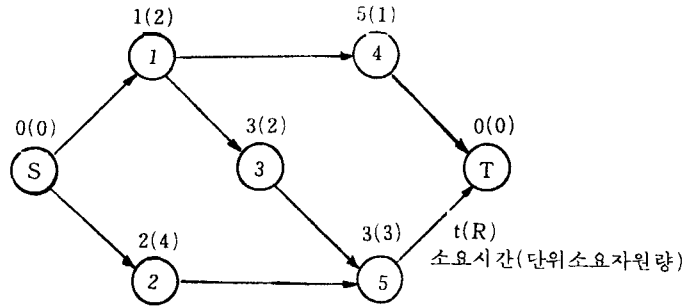


그림 1 수치예의 네트워크

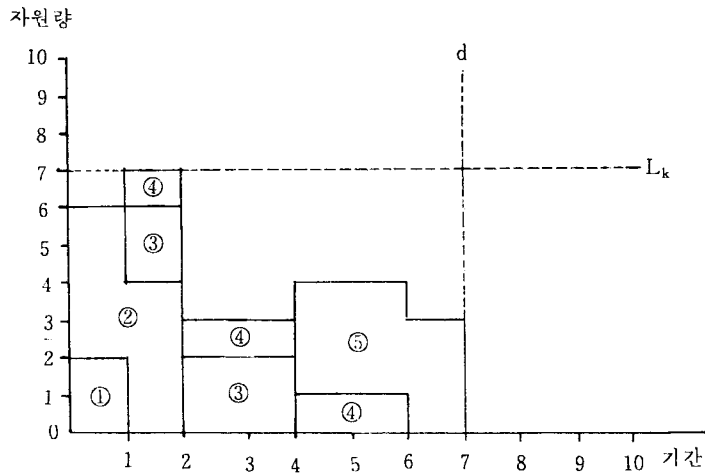


그림 2 수치예의 그래프

수치 예의 최적수준에 대한 그래프는 그림 3과 그림 4와 같다. 비용계수에 대한 최적수준은 제약조건을 초과하는 비용(잔업비용)과 남기지연에 따른 지연비용이 다른 비용계수보다 단위당 비용이 높은 비율이면 수준이 높게 설정된다(그림 3의 L=4이다).

만일 비용계수가 동일한 경우는 수준이 L=3이다. (그림4) 또 자원의 단위당 비용이 낮은 비율이면 수준이 높게 설정됨을 알 수 있다(표 1 참조).

표 1 비용계수에 대한 최적수준

수준	비용계수									
	W=10 V=10 A=10 B=10	W=20 V=10 A=15 B=8	W=5 V=10 A=10 B=10	W=20 V=10 A=10 B=10	W=10 V=5 A=10 B=10	W=10 V=20 A=10 B=10	W=10 V=10 A=5 B=10	W=10 V=10 A=20 B=10	W=10 V=10 A=10 B=5	W=10 V=10 A=10 B=20
L7	680	642	680	680	435	1170	680	680	585	870
L6	540	516	540	540	330	960	540	540	480	660
L5	480	497	465	510	305	830	475	490	435	570
L4	350	386*	335	380*	210*	630	340	350*	340	370
L3	320*	398	290*	380*	215	530*	300*	360	315*	330*

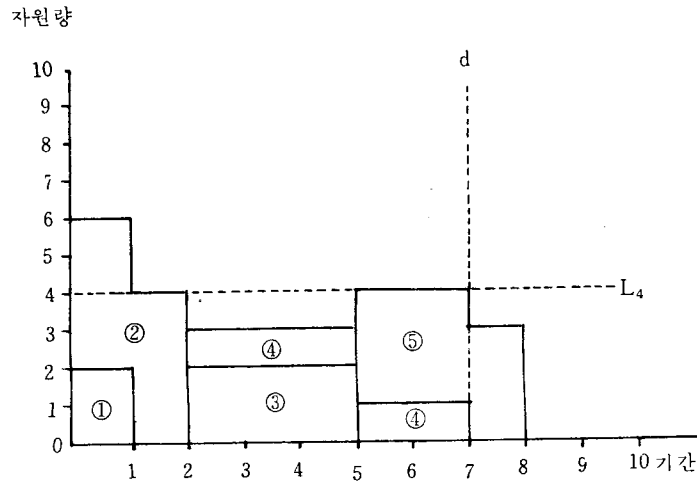


그림 3 L=4의 경우

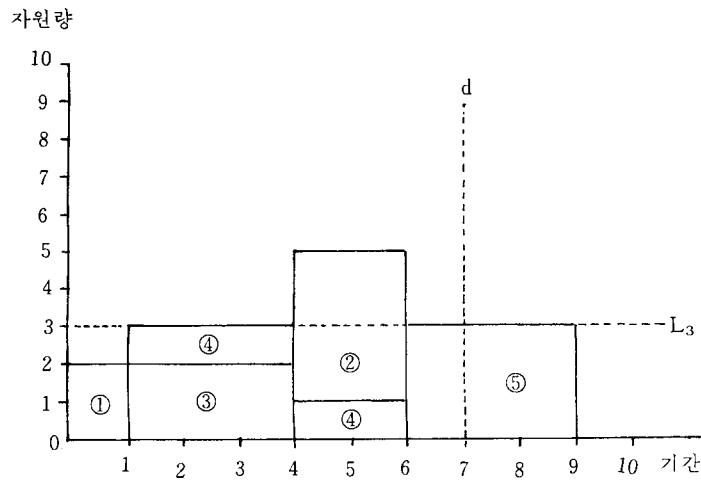


그림 4 L=3의 경우

5. 결 론

실제로 몇개의 수치실험을 해보았지만 잡이 n개의 경우 최초로 열거해야 하는 잡수에 있는 프로젝트 스케줄의 개수가 최대 $n!$ 이기 때문에 프로젝트 스케줄에 대해서 각 Job의 개시시간을 결정하는 것은 매우 큰 계산시간을 요한다.

또 자원의 수준을 결정할시에 자원수준을 미달할 경우를 고려하지 않으면 항상 사용가능한 자원량을 적게 설정하여 수준을 초과하는 잡수가 많게 되는 경향이 있다.

수치예의 각 비용계수의 비교에 의해서 제약수준의 초과비용, 남기지연비용 및 자원비용이 항상 사용가능한 자원량의 수준에 영향을 미친다는 것을 알 수 있다.

참 고 문 헌

1. Cooper D. F., "Heuristics for Scheduling Resource-Constrained Projects : An Experimental Investigation," *Management Science*, 22(11), 1976.

2. Davis E. W. and Patterson J. H., "A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling," *Management Science*, 21(8), 1975.
3. Fisher M. L., "Optimal Solution of Scheduling Problems Using Lagrange Multipliers : Part I," *Operations Research*, 21(5), 1973.
4. Mason A. T. and Modie C. L., "A Branch and Bound Algorithm for Minimizing Cost in Project Scheduling," *Management Science*, 18(4), 1971.
5. Mohanty R. P. and Siddio M. K., "Multiple Projects-Multiple Resources Constrained Scheduling : Some Studies," *Int. J. Prod. Res.*, 27(2), 1989.
6. Patterson J. H., "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource Project Scheduling Problem," *Management Science*, 30(7), 1984.
7. Smith-Daniels D. E. and Aguilano N. J., "Constrained Resource Project Scheduling Subject to Material Constraints," *Journal of Operations Management*, 4(4), 1984.
8. Stinson J. P., Davis E. W. and Khumawala B. M., "Multiple Resource-Constrained Scheduling Using Branch and Bound," *AIIE Transactions*, 10(3), 1978.
9. Willis R. J., "Critical Path Analysis and Resource Constrained Project Scheduling-Theory and Practice," *European Journal of Operational Research*, 21, 1985.
10. Yang T., Ignizio J. P. and Song J. S., "An Exchange Heuristic Algorithm for Project Scheduling with Limited Resources," *Eng. Opt.*, 14, 1989.