「經營情報硏究」 제5호
大韓經營情報學會
2000年 12月

비열등 프로젝트 일정 탐색

안태호* · 김명관** · 이동엽***

〈목 차〉

Ⅰ. 서 론

과제일정문제(Project Scheduling Problem)란 과제를 구성하는 활동(activity)의 집행시작시기(start time), 집행양식(mode) 및 집행기간(duration)을 선정하는 문제이다. 본고에서는 가용한 자원의 양이 한정되어 있으며 각 활동마다 과제담당자가 선택할 수 있는 집행양식이 다수이며 한 양식내에서 가능한 기간이 다수인 과제일정문제를 다룬다. 이러한 문제는 "자원의 양이 제약되고 기간단축이 가능한 복수의 양식을 지닌 단일 프로젝트 일정문제(Resource Constrained Project Scheduling Problem with Multiple Crashable Modes, RCPSPMCM)"라 불려지며, 이에 대한 연구문헌으로는 Ahn *et al.*(1995)과 안태호(1995)가 있다.

자원의 양이 제약된 과제문제(the Resource Constrained Project Scheduling Problem, 약칭 RCPSP)를 다루는 거의 모든 연구들은 "과제비용의 제약하의 과제완료

* 숭실대학교 경영학부 조교수
** 한국산업기술진흥협회 수석연구원
*** 혜천대학 정보시스템계열 조교수

시기 단축(Talbot, 1982, Patterson *et al.*, 1989, Sprecher, 1994)" 또는 "과제완료시기 제약하의 과제비용의 단축(Talbot, 1982, Patterson *et al.*, 1989, Ahn *et el.*, 1995, 안태호, 1995)"에 초점을 맞추어 왔다. 두 부류의 연구는 주어진 제약 하의 단일 목적최적화(비용 또는 완료시기단축)문제라 할 수 있다.

상기 두 부류의 연구와는 달리 본고에서는 다목적 문제를 풀고자 한다. 즉, 과제완료시기와 과제비용 모두를 줄이는 일정을 찾아내는 문제가 고려된다. 그런데 완료시기와 비용은 상호 역의 관계를 지니고 있어 완료시기를 단축하면 비용이 증가하고 비용을 감축하면 완료시기가 늘어나기 마련이다. 그러므로, 완료시기와 비용 모두를 줄이는 문제는 단일의 최적해를 지니지 않고 다수의 비열등해를 지니기 마련이다. 본고의 경우 완료시기와 비용을 줄이는 문제이므로, 열등해와 비열등해는 다음과 같이 정의될 수 있다. 두 개의 解, A와 B가 있다고 가정하고 해 A의 비용과 완료시기를 C_A와 T_A, 해 B의 비용과 완료시기를 C_B와 T_B라 하자. 만일 "$C_A < C_B$이고 $T_A \le T_B$"가 성립하거나 "$C_A \le C_B$이고 $T_A < T_B$"가 성립하면 "해 B는 해 A보다 열등하다"라고 한다. 만일 "$C_A < C_B$이고 $T_A > T_B$"가 성립하거나 "$C_A = C_B$이고 $T_A = T_B$"가 성립하면 "해 A와 해 B는 서로 비열등한 관계에 있다"라고 한다.

본고에서는 위에서 언급된 다목적문제를 수식화하고 최적해법을 소개하며 최적해법에 의한 모의계산결과를 보고하고자 한다.

Ⅱ. 수 식 화

과제는 $I > 0$개의 非가상(nondummy) 활동으로 이루어져 있다고 가정한다. 표기는 마디(node)가 활동을 의미하는 AON(activity on node)형식을 따랐다. 활동 0과 $(I+1)$은 과제의 시작과 종료를 나타내는 가상활동들이며, 가상활동들의 집행에는 시간이나 비용이 필요하지 않다. 문제의 수식화는 다음과 같다.

$$(P) \quad \text{Min} \quad Cost$$
$$\text{Min} \quad F_{I+1} \tag{1}$$

subject to

$$\text{Cost} = \sum_{i=1}^{I} \sum_{m \in M(i)} x_{i(m)} \times \{c_{i(m)n} + cr_{i(m)} \times (y_{i(m)n} - y_i)\} \tag{2}$$

$$x_{i(m)} \in \{0,1\}, \qquad \forall \; i, m \tag{3}$$

$$\sum_{m \in M(i)} x_{i(m)} = 1, \qquad \forall \; i, m \tag{4}$$

$$F_i + y_i \le F_j, \qquad (i,j) \in H \tag{5}$$

$$\sum_{i \in S_t} \sum_{m \in M(i)} x_{i(m)} r_{i(m)k} \le b_k, \qquad \forall \; k, t \tag{6}$$

$$x_{i(m)} y_{i(m)c} \le x_{i(m)} y_i \le x_{i(m)} y_{i(m)n}, \qquad \forall \; i, m \tag{7}$$

$$F_0 = 0 \tag{8}$$

$$y_0, y_{I+1} = 0, \tag{9}$$

여기서,

i: 활동, $i = 0, 1, \ldots, I+1$.

m: 활동의 양식

$M(i)$: 활동 i를 실행할 수 있는 양식의 집합, $m \in M(i)$

$x_{i(m)}$: 0-1 양식 변수;

$$x_{i(m)} = \begin{cases} 1, & \text{활동 i가 양식 m으로 집행되는 경우} \\ 0, & \text{o.w.} \end{cases}$$

y_i: 활동 i의 (집행)기간을 나타내는 변수

$c_{i(m)n}$ $(c_{i(m)c})$: 활동 i가 양식 m으로 집행되는 경우의 정상(단축)비용

$y_{i(m)n}$ $(y_{i(m)c})$: 활동 i가 양식 m으로 집행되는 경우의 정상(단축)기간

$cr_{i(m)}$: 활동 i가 양식 m으로 집행되는 경우의 한계단축비용

$$cr_{i(m)} = -\frac{c_{i(m)n} - c_{i(m)c}}{y_{i(m)n} - y_{i(m)c}}$$

F_i: 활동 i의 시작시기를 나타내는 변수

H: 선행관계를 나타내는 한 쌍의 활동들의 집합; $(i, j) \in H$라면 활동 j의 시작시기 (F_j)는 활동 i의 종료시기$(F_i + y_i)$보다 빠를 수 없음을 나타내며 활동 i는 활동 j의 선행활동이라 한다.

period: period t는 정수인덱스로서 시점 t-1에서 시점 t까지의 기간을 지칭한다. 시점 t-1은 포함되지 않고 시점 t는 포함된다.

$r_{i(m)k}$: 활동 i가 양식 m으로 집행되는 경우 활동 i가 집행되는 각 period 마다 활동 i에 투입되어야 할 자원 k의 양

K : 과제를 수행함에 있어 필요한 자원의 종류

b_k: 각 period마다 가용한 자원 k의 양

S_t : period t에 집행 중인 활동들의 집합;

$$S_t = \{ i \mid F_i < t \le F_j + y_j \}$$

활동 (I+1)은 과제의 완료를 나타내는 가상활동이며 활동 (I+1)의 집행기간은 0이므로, 활동 (I+1)의 시작시기를 나타내는 F_{I+1}은 과제의 완료시기가 된다. 과제에 투입되는 총비용은 과제를 이루고 있는 활동들에 투입되는 비용의 합으로서 수식 (2)에 나타나 있는데, 각각의 활동비용은 그 활동에 선정된 양식과 기간에 의해 결정된다. 임의의 비가상활동 i를 예로 들어보자. 활동 i를 실행할 수 있는 양식의 집합은 M(i)이다. 제약식 (3)과 (4)는 M(i)에 속한 양식 중 오직 하나만이 선정되어야 함을 나타낸다. 양식 m_i가 선정되었다고 가정하자. 제약식 (7)은 활동 i의 집행기간(y_i)은 선정된 양식의 정상기간($y_{i(m_i)n}$)과 단축기간($y_{i(m_i)c}$) 사이에 있어야 함을 나타낸다. 만약 y_i가 $y_{i(m_i)n}$과 동일하다면 활동 i의 집행에 소요되는 비용은 양식 m_i의 정상비용($c_{i(m_i)n}$)과 동일하다. 만약 y_i가 $y_{i(m_i)n}$보다 짧다면 기간단축이 시도된 것이며, 기간이 단축된 period마다 m_i의 한계단축비용인 $cr_{i(m_i)}$만큼 추가비용이 발생한다. 그러므로 활동 i가 양식 m_i로 집행되는 경우, 활동 i의 집행비용은 $c_{i(m_i)c} + cr_{i(m_i)} \times (y_{i(m_i)n} - y_i)$이며, 활동 i의 집행기간 y_i의 값은 $y_{i(m_i)c} \le y_i \le y_{i(m_i)n}$로 제한된다. 제약식 (5)는 활동들의 선행관계를 규정한다. 제약식 (6)은 자원의 제약에 관한 기술이다. 각 period에서 필요한 자원의 양은 해당 period에서 어떤 활동들이 어떤 양식으로 진행중(active)인가에 의해 결정된다. 임의의 기간 t($1 \le t \le F_{I+1}$)를 고려하자. 이 기간에 진행 중인 활동들은 u, v 및 w이며 해당 활동들의 선정된 양식은 각각 m_u, m_v 및 m_w라 가정하자. 이 경우 기간 t에 집행 중인 활동들에 투입되어야 할 자원 k의 양은 $r_{u(m_u)k} + r_{v(m_v)k} + r_{w(m_w)k}$이 된다. 제약식 (6)은 과제가 진행 중인 모든 기간에서 모든 자원별로 투입되어야 할 자원의 양이 가용한 자원의 양을 초과해서는 안됨을 나타

낸다. 그런데, 기간 t에 집행 중인 활동의 집합(S_t)은 각 활동들의 시작시기 및 기간에 의해 결정되므로, 제약식 (6)은 개념적 기술(conceptual statement)이라 불린다.

이 모델에서는 다음의 가정들이 전제되었다.

① 모든 $y_{i(m)n}$, $y_{i(m)c}$, $c_{i(m)n}$, $c_{i(m)c}$, $cr_{i(m)}$, $r_{i(m)k}$ 및 b_k의 값은 상수이며 그 값은 사전에 알려져 있다.

② 모든 $y_{i(m)n}$과 $y_{i(m)c}$는 정수이다.

③ 일단 시작된 활동은 종료되기 전까지 다른 활동에 의해 중단될 수 없다.

Ⅲ. 최적해법과 바운딩 규칙들

본고에서 소개될 다목적 일정문제 해법의 설명에는 RCPSP 분야의 전문용어가 많이 사용되므로, 필요한 RCPSP의 용어들을 먼저 설명하고자 한다(이러한 용어들의 자세한 설명은 Sprecher(1994)와 안태호(1995)에서 찾을 수 있다). 일정(schedule)이란 고려되는 문제에 따라 달리 정의된다. 각 활동마다 양식과 기간이 하나씩인 경우(RCPSP with a single mode) 일정이란 각 활동들의 "시작시기"로 구성된다. 각 활동마다 양식은 다수이나 각 양식마다 기간이 단일한 경우(RCPSP with multiple modes) 일정이란 각 활동들의 "시작시기 및 선정된 양식"으로 이루어진다. 각 활동마다 양식이 다수이고 양식 안에서도 기간이 복수인 경우(RCPSP with multiple crashable modes) 일정이란 각 활동들의 "시작시기, 양식 및 기간"으로 이루어져 있다. 만약 일정이 주어진 모든 제약조건을 만족하면 그 일정은 가능일정(feasible schedule)이라 불린다. 부분일정(partial schedule)이란 몇몇 활동들에 대하여 "시작시기(RCPSP with a single mode)," "시작시기 및 양식(RCPSP with multiple modes)" 또는 "시작시기 및 양식과 기간(RCPSP with multiple crashable modes)"이 배정된 경우를 지칭한다. 그러므로 부분일정에서 활동들은 해당 결정변수가 배정된 활동들과 아직 배정되지 않은 활동들로 구분된다. 임의의 부분일정이 다음의 두 조건을 만족시키면 그 부분일정은 부분가능일정(partial feasible schedule)이라 불린다: (1) 활동 i가 부분일정에서 배정되었다면, 활동 i의 모든 선행활동도 배정되었다. (2) 만약 과제가 이미 배정된 활동들만으로 이루어져 있다면, 즉 과제에서 아직 배정받지 못한 활동들을 제거한다면, 부분일정은 모든 제약조건을 충족시킨다.

본고에서 소개될 최적해법은 RCPSP 분야에서 가장 널리 알려진 특수해법 중의 하나인 역방향해법(Backtracking Algorithm)의 응용인데, 역방향해법은 Patterson *et al.*(1989)에서 처음으로 소개되었다. 본디 역방향해법은 RCPSP with multiple modes 문제의 해법으로 소개되었으나, 이 해법은 RCPSP with multiple crashable modes 문제에도 적용될 수 있으며 이에 필요한 수정절차는 Ahn *et al.*(1995)에 상술되어 있다. 역방향해법이란, "(활동들의)시작시기가 우월한 부분일정(start time dominating partial schedule, Patterson *et al.*, 1989 참조)" 모두를 체계적으로 탐색하는 절차(procedure)인데 탐색절차는 나무구조(tree structure)를 띠고 있다. 과제가 활동 0, 1, 2, ..., I+1로 이루어진 경우 나무구조의 단계(stage)는 총 (I+2)개가 된다. 나무구조는 활동들의 배정순서를 나타낸다. 나무구조는 단계 1부터 시작하며, 나무구조에서 아래 단계로(단계 j-1에서 단계 j) 전개해 나갈 때에는 아직 배정받지 못한 활동이 그 단계에 배정되는데, 특히 활동의 시작시기는 다음의 세 조건을 충족시키는 시작시기 중 가장 빠른 것을 택한다. (1) 현 단계에서 배정될 활동과 이미 배정된 활동들은 어떤 기간에서도 자원제약(제약식 6)을 위반하지 않는다. (2) 현 단계에서 배정될 활동은 모든 선행관계(제약식 5)를 위반하지 않는다. (3) 현 단계에서 배정될 활동의 시작시기는 이미 배정된 어떤 활동의 시작시기보다 빠르지 않다. 마지막 조건 (3)은 역방향해법에서 precedence tree 제약이라 불리며, 이 조건의 필요성에 대하여는 Patterson *et al.*(1989)에 상술되어 있다. 나무구조의 j단계에서 (j-1)단계로 올라갈 때는 (backtrack) 단계 j에서 배정되었던 활동은 미배정상태로 변한다. 그러므로, 해법이 나무구조 j단계에 있다면 단계 j로부터 단계 1까지 배정된 활동들의 배정값을 추적함으로써 j개의 활동으로 이루어진 부분일정을 도출할 수 있다. 그러므로 일정은 단계 (I+2)에서만 작성된다. 또한 나무구조의 특성상 j단계에서 j-1단계로 가는 가지(branch)는 하나이나 j-1단계에서 j단계로 가는 가지는 복수일 수 있다. 이는 j (j<I+2)단계에서 최종단계 (I+2)로 가는 가지가 복수일 수 있음을 의미하므로 임의의 부분가능일정에서 전개될 수 있는 가능일정 또한 복수일 수 있다.

과제비용과 과제완료시기의 측면에서 비열등한 일정의 집합은 시작시기가 우월한 일정의 집합의 부분집합이므로 역방향해법에 의해 발견되는 일정들을 조사하면 우리가 찾고자 하는 비열등한 일정의 집합을 구할 수 있다. 그러나 과제를 구성하고 있는 활동의 수, 각 활동마다 가능한 양식의 수, 그리고 한 양식에서 가능한 기간의 수가 늘어날수록 역방향해법에 의해 발견되는 일정의 수 또한 기하급수적으로 증가하므로 본고에서 제기된 문제를 역방향해법만을 이용하여 해결한다는 것은 거의 불가능하다. 이러한 문제점을 개선하기 위하여 다음의 바운딩 규칙(bounding rule)들을 도입하는데, 바운딩규칙

(bounding rule)이란 해법의 효율성을 높이기 위하여 도입된 규칙을 말하며 현재 조사하고 있는 부분가능일정에서 전개되는 가능일정들을 탐색할 필요가 없다고 판단되면 그 부분일정을 더이상 탐색하지 않고 나무구조에서 역방향으로 움직이도록(backtrack) 해준다.

1. 비열등일정 바운딩 규칙

비열등일정 바운딩 규칙은 현재 조사중인 부분가능일정으로부터 전개하여 얻을 수 있는 어떤 가능일정도 이미 알려진 가능일정보다 열등한 일정일 경우 그러한 부분가능일정은 더 이상 전개할 필요가 없다는 것이다.

1.1 정리 1

임의의 부분가능일정 A를 가정하자. 부분가능일정 A로부터 전개하여 얻을 수 있는 모든 가능일정의 최소완료시기와 최소비용을 부분가능일정 A의 최소완료시기(T_A)와 최소비용(C_A)라 하고, 이미 알려져 있는 임의의 가능일정 B의 완료시기와 비용을 T_B와 C_B라 하자. 만일 "$C_A > C_B$이고 $T_A \geq T_B$"가 성립하거나 "$C_A \geq C_B$이고 $T_A > T_B$"가 성립하면 부분가능일정 A로부터 전개하여 얻어지는 어떤 가능일정도 비열등 일정일 수 없으므로, 부분가능일정 A는 더 이상 전개할 필요가 없다.

[증명] 일정 C는 부분일정 A로부터 전개하여 얻어지는 가능일정 중 임의로 선택된 일정이라 하고, 일정 C의 실제비용과 실제완료시기를 C_C 및 T_C라 하자. C_A와 T_A의 정의로부터 $C_C \geq C_A$이며 $T_C \geq T_A$가 성립한다. 만일 "$C_A > C_B$이고 $T_A \geq T_B$"가 성립하면 "$C_C > C_B$이고 $T_C \geq T_B$" 또한 성립하고, "$C_A \geq C_B$이고 $T_A > T_B$"가 성립하면 "$C_C \geq C_B$이고 $T_C > T_B$" 또한 성립하므로 일정 C는 일정 B보다 열등하다. 일정 C는 부분일정 A로부터 전개하여 얻어지는 가능일정 중 임의로 선택되었으므로, 이는 부분일정 A로부터 어떤 비열등일정도 전개될 수 없음을 보인다.

1.2 비열등일정 바운딩 규칙에 따른 해법의 수정

비열등일정 바운딩 규칙을 사용하기 위하여는 임의의 부분가능일정의 최소완료시기와 최소비용을 산출하여야 한다. 최소완료시기와 최소비용을 산출하기 위해서는 모든

활동들의 기간과 비용이 산정되어야 하므로, 아직 배정받지 못한 활동들의 경우 그 활동이 지닌 "양식과 기간의 조합" 중 가장 짧은 기간을 그 활동의 최소기간으로 하고 조합 중 가장 저렴한 비용을 그 활동의 최소비용으로 삼는다. 부분일정의 최소완료시기는 아직 배정받지 못한 활동들에게 임시로 시작시기 및 기간을 배정함으로써 구해지는데, 임시 시작시기는 활동들의 선행관계(precedence relation)를 만족시켜야 하고 이미 배정받은 어떤 활동들의 시작시기보다는 빠를 수 없으며, 임시기간은 그 활동의 최소기간으로 한다. 부분일정의 최소완료시기는 과제의 완료를 나타내는 가상활동 (I+1)의 종료시기가 되며, 활동 (I+1)의 기간은 0이므로 활동 (I+1)의 종료시기는 시작시기와 동일하다. 부분일정의 최소비용은 이미 배정받은 활동들의 비용의 합에 아직 배정받지 못한 활동들의 최소비용을 더하여 산정한다.

정리 1은 이미 알려져 있는 가능일정의 실제완료시기와 실제비용을 필요로 하므로, 정리 1을 사용하기 위하여 ESS(the efficient schedule set)를 "지금까지 발견된 일정 중 비열등한 일정의 집합"이라 하자. 해법이 시작될 때 ESS는 공집합으로 한다. 역방향해법을 진행 중 ESS에 있는 일정보다 비열등한 가능일정이 작성되면 ESS는 그 가능일정을 포함하게 되며, ESS 안에 있는 가능일정 중 새로이 포함된 가능일정보다 열등한 일정이 있으면 그러한 열등한 일정은 ESS에서 삭제된다. 다시 말해서 ESS는 공집합으로 시작하여 해법이 진행되는 동안 일정의 완료시기와 비용이라는 면에서 비열등한 일정들을 수록하게 되며 역방향해법이 종료되었을 때는 "모든 가능일정 중 비열등한 일정의 집합"이 된다.

이제 정리 1을 사용하기 위한 모든 수정절차가 완료되었으므로, 역방향해법이 새로운 부분가능일정을 작성할 때마다 그 부분가능일정의 최소완료시기와 최소비용을 산출한 후 ESS 안에 있는 가능일정들과 비교하여 작성된 부분일정이 열등하다 판명되면 그 부분일정은 더 이상 전개하지 않는다.

2. 기존 RCPSP 바운딩 규칙의 적용

기존 RCPSP 연구분야에 알려져 있는 바운딩 규칙들 중 다음의 바운딩 규칙은 본고에서 고려되고 있는 문제에도 응용될 수 있다. 이 규칙들의 자세한 내용은 Sprecher (1994)와 안태호(1995)에 상술되어 있으므로, 본고에서는 바운딩 규칙의 자세한 설명보다는 이러한 바운딩 규칙을 본고에서 제기된 문제에도 적용시킬 수 있음을 보이는 데 초점을 맞추었다.

2.1 Sprecher(1994)의 바운딩 규칙 2 및 4의 적용

역방향해법은 동일한 부분가능일정도 만약 활동의 배정순서(각 활동이 배정된 단계)가 다르다면 상이한 부분가능일정으로 여긴다. 부분가능일정 A와 B가 배정순서만 다를 뿐 나머지는 동일하다고 가정하자. 역방향해법의 특성상 위의 가정이 성립하면, 부분가능일정 A에서 전개되는 어떤 가능일정도 배정순서만 다를 뿐 부분가능일정 B로부터 전개된다. 그러므로 부분가능일정 A가 검토되었다면 부분가능일정 B는 검토 및 전개될 필요가 없다. Sprecher의 바운딩 규칙 2는 역방향해법이 배정순서만 다를 뿐인 부분가능일정을 반복하여 전개하는 것을 방지한다. 이미 조사된 부분가능일정은 다시 검토할 필요가 없으므로, 본고에서도 Sprecher의 바운딩 규칙 2를 사용하였다. 그러나, Sprecher의 바운딩 규칙은 한 양식내에서 가능한 기간이 하나인 경우(RCPSP with multiple modes)를 상정하였고 본고에서는 한 양식내에서 가능한 기간이 복수인 경우(RCPSP with multiple crashable modes)를 고려하였으므로, Sprecher의 바운딩 규칙 2에서 사용된 "활동, 활동의 배정순서, 양식"으로 이루어진 3차원의 array는 본고에서는 "활동, 활동의 배정순서, 양식, 기간"으로 구성된 4차원의 array로 수정되었다.

Sprecher의 바운딩 규칙 4는 Demeulemeester *et al.*(1992)에 의해 고안된 left-shift 규칙의 응용이다. left-shift 규칙을 설명하기 위하여 다음의 두 부분일정 A와 B를 고려하자. "부분일정 A에서 전개될 수 있는 가능일정 중 임의의 일정을 C라하면, 부분일정 B에서도 일정 C와 유사한 일정 D가 반드시 전개되는데 일정 C와 일정 D의 차이점은 오직 한 활동의 배정 시작시기가 일정 D에서 빠르다." 위의 조건이 성립하게 되면 일정 C는 일정 D에 비하여 "활동들의 시작시기 면에서" 열등하다고 하며 부분일정 A 또한 부분일정 B에 대하여 "활동들의 시작시기 면에서" 열등하다고 하는데, left-shift 규칙이란 위의 조건의 성립 여부를 효과적으로 검색하는 기법이다. 일정 C와 일정 D의 차이점은 오직 한 활동의 배정 시작시기이므로, 일정 C와 일정 D의 비용은 동일하다. 그러므로, 본고에서 고려되는 문제에서도 "활동들의 시작시기 면에서" 열등한 일정이나 부분일정은 검색할 필요가 없으므로 left-shift 규칙을 사용하였다.

2.2 안태호(1995)의 바운딩 규칙 1 및 2의 적용

안태호의 바운딩 규칙 1을 설명하기 위하여 부분일정 A와 B를 고려하자. "부분일정 A에서 전개될 수 있는 가능일정 중 임의의 일정을 C라면, 부분일정 B에서도 일정 C와 유사한 일정 D가 반드시 전개되는데 일정 C와 일정 D의 차이점은 오직 한 활동 j의 배

정값(시작시기, 양식, 기간)이 다르며 일정 C에서의 활동 j의 비용보다 일정 D에서의 활동 j의 비용이 적다." 과제의 종료를 나타내는 가상활동 (I+1)은 어떤 비용도 발생시키지 않으므로, 위에서 언급된 활동 j일 수 없다. 활동 (I+1)의 기간은 0이므로, 과제의 종료시기는 바로 활동 (I+1)의 시작시기이다. 그러므로, 일정 C와 일정 D는 동일한 과제완료시기를 갖는다. 일정 C와 일정 D에서 오직 활동 j의 배정값이 다르며 일정 C에서의 활동 j의 비용이 일정 D에서의 활동 j의 비용보다 많으므로, 일정 C는 일정 D에 비하여 "과제비용 면에서" 열등한 일정이다. 일정 C는 부분일정 A에서 전개된 일정중 임의로 택한 일정이므로, 위의 조건이 성립하면 부분일정 A는 부분일정 B에 비하여 "과제비용 면에서" 열등한 부분일정이 된다. 안태호의 바운딩 규칙 1은 "과제비용 면에서" 열등한 부분일정을 효과적으로 탐색하는 기법이며, 본고에서 고려된 문제에서도 "과제비용 면에서" 열등한 부분일정은 고려할 필요가 없으므로 이 바운딩 규칙 또한 사용되었다.

안태호의 바운딩 규칙 1은 한 활동의 배정값을 변화시켜 현재 고려중인 부분일정보다 "과제비용 면에서" 우월한 부분일정을 작성할 수 있는가를 조사하는 기법이라면, 안태호의 바운딩 규칙 2는 두 활동의 배정값을 변화시켜 현재 고려 중인 부분일정보다 "과제비용 면에서" 우월한 부분일정을 작성할 수 있는가를 조사하는 기법이다. 본고에서 고려된 문제에서도 "과제비용 면에서" 열등한 부분일정은 고려할 필요가 없으므로 이 바운딩 규칙 또한 사용되었다.

Ⅳ. 모의계산의 결과

본고에서 소개된 해법의 효율성을 조사하기 위하여 (1) 해법을 C언어로 프로그램화하였으며 (2) 모의문제를 작성하였고 (3) 작성된 프로그램을 이용하여 모의문제의 비열등 가능일정 집합을 구하는데 소비된 CPU 시간을 조사하였다. 계산에는 Pentium 60Mhz(IBM compatible, 16MB 주메모리)의 PC가 사용되었다. 총 8개 set의 문제가 다루었는데, 각 set은 10문제로 이루어져 있다. 사용된 문제는 단일목적함수문제를 다룬 Ahn *et al.*(1995)에서 사용된 문제와 동일하다. 사용된 문제들은 과제의 활동수, 각 활동당 양식의 수, 각 양식당 가능한 기간의 수에 따른 해법의 효율성을 검증하기 위하여 Ceteris Paribus 디자인을 따르고 있다. 표 1은 총 8 set의 문제특성과 해당 set의 문제를 푸는데 소비된 CPU 시간 측정결과를 나타내고 있다. Ceteris

Paribus 디자인의 베이스는 Set 1이며, Set 1에 포함된 문제들은 "비가상활동의 수가 10, 활동당 가능한 양식의 수가 2, 한 양식내에 가능한 기간의 수가 3, Complexity of Network(약칭 C, 과제네트워크에서 쓰이는 용어로, 총 선행관계의 수를 총활동의 수로 나눈 값이다.)는 1.8"이다. Set 1의 문제에 대한 자세한 설명은 Ahn *et al.* (1995)에 상술되어 있다. Set 2부터 Set 8까지는 Set 1에 사용된 parameter의 값을 변화시킴으로써 구해지는데, 변화된 parameter의 값은 수정칸에 설명되어 있다.

〈표 1〉 모의계산결과

Set 번호	수 정	CPU 시간측정(단위 초)	
		평균 〔표준편차〕	최대 〔최소〕
1	없음	20.10 〔26.71〕	97.00 〔2.00〕
2	과제의 활동의 수가 2 증가	105.60 〔70.65〕	266.00 〔33.00〕
3	과제의 활동의 수가 4 증가	544.20 〔335.18〕	1,157.00 〔116.00〕
4	C가 0.3 감소	24.60 〔8.43〕	35.00 〔9.00〕
5	C가 0.3 증가	13.30 〔15.42〕	49.00 〔3.00〕
6	양식당 가능한 기간이 1 증가	65.20 〔62.66〕	213.00 〔7.00〕
7	활동 당 양식의 수가 1 감소	1.70 〔2.24〕	8.00 〔0.00〕
8	활동 당 양식의 수가 1 증가	95.40 〔59.08〕	202.00 〔13.00〕

V. 결 론

지금까지의 RCPSP 분야의 연구는 주로 제약하의 단일목적문제에 초점이 맞추어져 왔다. 본고에서는 RCPSP with multiple crashable modes의 다목적문제를 다루었는데, 목적함수는 과제의 완료시기와 과제비용의 동시 단축이다. 완료시기와 비용은 상

호 역의 관계를 지니므로, 본고에서 고려된 문제는 다수의 비열등해를 찾는 문제가 된다. 본고에서 소개된 최적해법은 역방향해법에 기초하며, 새로운 바운딩 규칙과 종래 RCPSP 분야에서 사용된 규칙들을 포함하고 있다. 소개된 최적해법의 효율성은 모의계산결과를 통하여 보고되었다.

참 고 문 헌

1. 안태호(1995), "자원의 양이 제약되고 기간단축이 가능한 복수의 양식을 지닌 단일 프로젝트 일정문제-자원의 용량이 시간에 따라 변하는 경우의 최적해법," 숭실경영경제연구 사회과학편 제25집, 103-119.
2. Ahn, T. and Selcuk, S.(1995), "Resource Constrained Project Scheduling Problem with Multiple Crashable Modes," *Working Paper Series*, Publication #95-101, DIS, CBA, University of Florida, Gainesville, FL 32611
3. Demeulemeester, E.L. and Herroelen, W.S.(1992), "A Branch and Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem," *Management Science* 38, 1803-1818.
4. Patterson, J.H., Slowinski, R., Talbot, F.B. and Weglarz, J.(1989), "An Algorithm for a General Class of Precedence and Resource Constrained Scheduling Problems," in: R. Slowinski and J. Weglarz (ed.), Studies in Production and Engineering Economics, vol. 9 : Advances in Project Scheduling, Elsevier, Amsterdam, 3-28.
5. Sprecher, A.(1994), "Resource-Constrained Project Scheduling : Exact Methods for the Multi-Mode Case," *Lecture Notes in Economics and Mathematical Systems* 409, Springer-Verlag, Berlin.
6. Talbot, F.B.(1992), "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs : The Nonpreemptive Case," *Management Science* 28, 1197-1210.

Abstract

An Exact Solution Method for Finding Nondominated Project Schedules

Ahn, Tae-ho · Kim, Myung-kwan · Lee, Dong-yeup

Project Managers want to reduce the cost and the completion time of the project simultaneously. But the project completion time tends to increase as the project cost is reduced, and the project cost has a tendency to increase as the project completion time is reduced. In this paper, the resource constrained project scheduling problem with multiple crashable modes is considered. An exact solution method for finding the efficient solution set and the computational results are introduced.