

자원제약을 고려하며 기간단축이 가능한 복수의 양식을 지닌
단일 프로젝트의 일정문제 : 자원 가용량이 시간에 따라
변하는 경우의 휴리스틱 기법

안 태 호*

Resource Constrained Project Scheduling Problem with
Multiple Crashable Modes: A Heuristic Procedure for the
Resource Availabilities Varying from Period to Period

Tae ho Ahn*

Abstract

In this paper a heuristic procedure for a resource constrained project scheduling problem with multiple crashable modes is presented. A similar heuristic procedure by Ahn and Erenguc was recently introduced, but the procedure assumes the resource capacities constant over the project life. The procedure of this paper is able to deal with the resource capacities varying over the project life. The computational results with a set of 110 test problems demonstrate the efficacy of the heuristic procedure.

* 숭실대학교 경영학부 조교수
논문접수: 98.11.16. 심사완료: 98.12.17.

I. 서론

프로젝트 일정문제(Project Scheduling Problem)는 프로젝트를 이루고 있는 활동(Activity)들의 집행에 관한 의사결정 문제이다. 프로젝트 일정기법이 효율적 프로젝트 관리에 큰 공헌을 할 수 있음에도 불구하고 일정기법이 현업에서 외면 당하는 주요 원인으로서는 지금까지 보편화된 일정기법들의 가정이 비현실적이며 프로젝트 관리자가 지닌 많은 대안들을 고려치 못하며 또한 프로젝트 관리의 주요 관심사인 자원관리를 적절히 통합하지 못하는 데 기인한다고 할 수 있다. 근자에 들어 이러한 문제점들을 보완하는 방향으로 많은 연구가 진행되고 있는 것은 무척 고무적인 일이다. 프로젝트 일정문제에 자원관리를 통합시킨 연구는 "자원제약을 고려한 프로젝트 일정문제 (Resource Constrained Project Scheduling Problem, RCPSP)"라 불리운다. 자원제약이란 프로젝트 수행에 동원되는 자원 (주요 설비, 인원, 장비, 예산 등)들이 프로젝트 기간 중 무한하지 않다는 인식이다. RCPSP란 각 기간 별로 프로젝트에 의해 요구되는 자원의 양이 가용한 자원의 양을 초과하지 않도록 일정을 작성하는 데 특색이 있다. RCPSP에서의 핵심용어는 양식(mode)으로서, 양식이란 어떤 활동을 수행하는 데 동원되는 자원과 비용 그리고 기간의 조합을 의미한다. 예를 들어 활동 A를 수행하는 방법으로는 설비 X 2대와 고급 인력 1명을 10일 동안 투입하여 총 200의 비용으로 수행하거나 (양식 1) 설비 Y 3대와 중급 인력 3명을 12일 동안 투입하여 총 250의 비용으로 수행할 수 있다고 하자 (양식 2). 비용만을 감안하면 양식 1이 양식 2보다 바람직하나 설비 X 또는 고급 인력이 여러 활동에 의해 동시에 요구되는 상황에서는 프로젝트의 지연을 방지하기 위해 양식 2를 선택하여야 하는 경우도 발생한다. "활동마다 단일의 양식이 존재하는 경우의 일정문제"란 각 활동의 시작시기를 결정하는 문제다 [5, 6, 7, 10]. 각 활동마다 복수의 양식이 존재할 수 있다는 인식에서 "각 활동의 양식이 복수인 경우의 문제 (RCPSP with Multiple Modes, RCPSPMM)"가 제기되었고 이 문제

에서 프로젝트 일정이란 활동 별 시작시기 및 양식을 의미한다 [8, 9, 11, 12]. RCPSP 분야에서 가장 최근에 소개된 일정문제는 "활동마다 하나 이상의 양식이 존재하며 양식 내에서 기간단축이 가능한 경우 (RCPSP with Multiple Crashable Modes, RCPSPMCM)"이다. RCPSPMCM의 주요 개념은 "기간단축이 가능한 양식 (Crashable Mode)"으로 위의 예를 사용하여 설명하여 보자. 활동 A를 양식 1로서 집행할 때 하루 8시간씩 작업하면 작업기간이 10일 (총 80시간)로서 비용은 200이다. 양식, 즉 설비 X 2대와 고급인력 1명이란 조합은 변경시키지 않고 하루 10시간씩 작업한다면 작업기간은 8일로 단축되는 반면 초과근무(overtime)비용이 발생하여 인건비는 상승하게 된다. 이 문제에서 작업기간 및 비용은 선정된 양식과 그 양식 내에서의 기간단축의 정도에 따라 결정되며 이 문제에서의 프로젝트 일정이란 각 활동 별 양식, 집행기간 및 시작시기를 의미한다. 이 부류의 문제는 프로젝트 일정문제의 두 부류, RCPSP와 기간/비용 상관문제(Time/Cost Trade-off Problem),의 통합이라는 의미를 지니고 있으며 Ahn [2]에 의하여 최초로 소개되었다. 각 자원의 가용량(Availability)이 프로젝트 기간 중 변하지 않는 경우의 최적해법은 Ahn and Erenguc [3]에 의해서, 동일 문제에 대한 휴리스틱 기법은 Ahn and Erenguc[4]에 의해 소개되었다. 프로젝트 기간 중 자원(들)의 가용량이 프로젝트 기간 중에 변하는 경우의 최적해법은 안태호[1]에 의해 연구되었으나, 이에 관한 휴리스틱 기법은 아직 소개된 바 없다. RCPSPMCM의 특성상 프로젝트를 이루고 있는 활동의 수, 활동 내의 양식의 수, 양식 내의 가능한 기간이 늘어남에 따라 최적해를 탐색하는 데 소요되는 시간은 기하급수적으로 증가하므로, 짧은 시간 내에 비교적 좋은 일정을 작성하는 휴리스틱 기법의 개발은 의의가 크다 할 것이다. 본고에서는 문제의 수학적 모형을 소개하고, Ahn and Erenguc[4]의 휴리스틱 기법을 변형하여 본 문제에 적합한 수정 휴리스틱 기법을 제시하며, 모의실험을 통하여 수정 휴리스틱 기법의 효율성을 검증하고자 한다.

II. 수학적 모형

프로젝트는 $J > 0$ 개의 활동들로 이루어져 있고 비음의 만기일($\bar{T} > 0$)이 있다고 가정한다. 표기는 마디(node)가 활동을 의미하는 activity-on-node형식을 따랐다. 활동 1과 J은 프로젝트의 시작과 종료를 나타내는 가상(dummy)활동들이다. 문제의 수식화는 다음과 같다.

$$(P) \min \left[\sum_{j=2}^{J-1} \sum_{m_j=1}^{M_j} x_{jm_j} \times NC_{jm_j} + CR_{jm_j} \times t \{ ND_{jm_j} - d_j \} + \bar{P}z \right] \dots\dots\dots(1)$$

subject to

$$x_{jm_j} \in \{0,1\}, \quad \forall j, m_j(2)$$

$$\sum_{m_j=1}^{M_j} x_{jm_j} = 1, \quad \forall j \quad \dots\dots\dots(3)$$

$$f_i \leq f_j - d_j, \quad i = 1, \dots, J-1, j \in S_i, \dots\dots\dots(4)$$

$$\sum_{i \in SA_p} \sum_{m_i=1}^{M_i} r_{jm_i,k} x_{jm_i} \leq R_p^k, \quad \forall k, p \dots\dots\dots(5)$$

$$f_1 = 0, \dots\dots\dots(6)$$

$$CD_{jm_j} x_{jm_j} \leq d_j x_{jm_j} \leq ND_{jm_j} x_{jm_j}, \quad \forall j, m_j \dots\dots\dots(7)$$

$$z \geq 0, \quad \dots\dots\dots(8)$$

$$z \geq f_J - \bar{T}, \quad \dots\dots\dots(9)$$

$$d_1, d_J = 0, \quad \dots\dots\dots(10)$$

여기서,

j : 활동을 나타냄, j = 1, 2, ..., J

Mj : 활동 j를 실행할 수 있는 양식의 수

mj : 활동의 집행양식

period: period p는 정수로서 시점 p-1에서 시점 p까지의 기간을 지칭한다. (시점 p-1은 포함되지 않고 시점 p는 포함된다.)

x_{jm_j} : 0-1 양식변수

$$x_{jm_j} = \begin{cases} 1, & \text{활동 } j \text{가 양식 } m_j \text{로 집행되는 경우} \\ 0, & \text{기타} \end{cases}$$

dj : 활동 j의 (집행)기간을 나타내는 변수

fj : 활동 j의 시작시기를 나타내는 변수

ND_{jm_j} : 활동 j가 양식 mj로 집행될 때 정상기간(normal duration)로 가장 긴 집행기간

CD_{jm_j} : 활동 j가 양식 mj로 집행될 때 단축기간(crash duration)으로 가장 짧은 집행기간

NC_{jm_j} / CC_{jm_j} : 활동 j가 양식 mj로 집행될 때 정상/단축비용(normal / crash cost)

CR_{jm_j} : 활동 j가 양식 mj으로 집행될 때 한계단축비용(crashing cost per period)

$$CR_{jm_j} = - \frac{NC_{jm_j} - CC_{jm_j}}{ND_{jm_j} - CD_{jm_j}}$$

Sj : 활동 j의 직접 후행활동들(immediate successors)의 집합

K : 프로젝트를 수행함에 있어 필요한 자원의 종류

$r_{jm,k}$: 활동 j가 양식 mj으로 집행될 경우 매 기간마다 필요한 자원 k의 양

R_p^k : period p에서 가용한 자원 k의 양

SAp : period p에서 집행 중인 활동들의 집합

$$SAp = \{ j \mid f_j - d_j < p \leq f_j \}$$

\bar{T} : 프로젝트의 만기일(deadline)

\bar{P} : 프로젝트가 만기일 이후까지 진행되는 경우 각 기간마다 부과되는 범칙금

z : 프로젝트의 지연정도(tardiness)를 나타내는 변수

$$z = \max \{ 0, f_J - \bar{T} \}$$

본고에서는 활동비용의 합계와 범칙금으로 이루어진 프로젝트 총비용을 최소화하는 문제를 다룬다. 활동비용은 선정된 양식과 기간에 의해 결정된다. 임의의 활동 j를 예로 들어보자. 활동 j를 실행할 수 있는 양식의 수는 M_j 이다. 제약식 (2)와 (3)은 활동 j가 {1, ..., M_j } 중 오직 하나의 양식으로 집행되어야 함을 나타낸다. 양식 \bar{m}_j 이 선정되었다고 가정하자. 제약식 (7)은 d_j 가 $ND_{j\bar{m}_j}$ 와 $CD_{j\bar{m}_j}$ 사이에 있어야 함을 나타낸다. d_j 가 $ND_{j\bar{m}_j}$ 이라면 이때의 활동집행비용은 $NC_{j\bar{m}_j}$ 가 된다. d_j 가 $ND_{j\bar{m}_j}$ 보다 짧은 경우 기간단축(crashing)이 시도된 것이며, 단축된 매 기간마다 $CR_{j\bar{m}_j}$ 만큼의 추가비용이 발생한다. 그러므로, 추가비용의 합은 목적식 (1)

에 나타나 있듯 $CR_{jm} \times (ND_{jm} - dj)$ 이고 활동의 집행비용은 NC_{jm} 를 더한 값이다. 본고에서 프로젝트는 이미 알려진 프로젝트 마감일(\bar{T})을 준수하여야 한다. 그렇지 않은 경우 지연된 매 기간마다 일정한 범칙금(\bar{P})이 부과되는 데, 프로젝트의 지연기간은 제약식 (8)과 (9)에 나타나 있듯 변수 z 로 표기되며 지연으로 인한 범칙금은 $\bar{P} \times z$ 이다. 목적식 (1)은 프로젝트 총비용의 최소화를 나타낸다. 제약식 (4)는 활동들간의 선행관계(precedence relation)를 보이는 것으로, 모든 활동은 해당 선행활동(preceding activity)들이 종료된 이후에나 시작될 수 있음을 나타낸다. 제약식 (5)는 자원의 용량에 관한 개념적 기술(conceptual statement)이다. 일정에 따라 프로젝트의 각 기간마다 진행 중인 활동들의 집합(SAp, $p = 1, \dots, fj$)이 결정되며, 제약식 (5)는 각 자원별로 각 기간에서 집행 중인 활동들에 의해 요구되는 자원의 양($\sum_{i \in SA_p} \sum_{m_i=1}^{M_i} r_{jm,k} x_{jm}$)은 그 기간에서 가용한 자원의 양(R_p^k)을 초과할 수 없음을 의미한다. 이 모형에서는 다음의 가정들이 전제되었다.

1. 모든 계수들, $ND_{jm}, CD_{jm}, NC_{jm}, CR_{jm}, r_{jm,k}, R_p^k, \bar{T}$ 및 \bar{P} 의 값은 사전에 알려져 있으며 불변이다.
2. 한 활동이 집행되기 시작하면 종료할 때까지 다른 활동에 의해 중단될 수 없다(nonpreemptive).
3. 활동의 집행에 있어 집행양식이 도중에 변경될 수 없다.

III. 수정 휴리스틱 기법

본고에서 제시하는 수정 휴리스틱 기법은 프로젝트 기간 중 자원(들)의 가용량이 기간별로 일정한 경우를 가정한 Ahn and Erenguc[4]의 휴리스틱 기법(이하 A&E 기법)을 기간별로 가용한 자원(들)의 가용량이 일정치 않은 경우에 적용하기 위해 수정한 기법이다. 수정기법의 제시요령은 다음과 같다. 수정기법과 A&E가 동일한 부

분은 A&E 기법이 참고문헌 [4]에 상세히 설명되었으므로 약술하고 수정 및 변형된 부분만 자세히 설명하고자 한다.

수정 및 A&E 기법은 모두 두 단계(stages)로 이루어져 있다. 첫 번째 단계에서는 프로젝트의 모든 제약조건을 충족시키는 실행가능한 일정(feasible schedule)을 작성한다. 두 번째 단계에서는 실행가능한 일정을 개선시키고자 6개의 개선규칙을 반복적으로 사용한다. (여기서 개선이란 일정의 실행가능성은 유지한 채 프로젝트의 총비용을 감소시키는 것을 의미한다.) 6개의 개선규칙으로 더 이상 일정이 개선되지 않으면 기법은 다시 첫 번째 단계로 옮겨간다. 이 과정은 컴퓨터 실행시간이 이미 설정한 제한시간을 경과할 때까지 반복된다.

1. 첫 번째 경로(Pass)에서의 최초 실행가능한 일정 작성

이 단계는 A&E 기법과 수정 기법은 동일하다. 이 단계에서 프로젝트 일정은 다시 두 단계로 작성된다. 첫 번째 단계에서는 각 활동들의 양식과 기간을 선정하고 둘째 단계에서는 활동들의 시작시기를 설정한다. 각 활동의 양식과 기간은 집행비용이 가장 저렴하도록 선정한다. (이는 프로젝트 총비용이 각 활동들의 집행비용의 합과 프로젝트 만기일을 지키지 못할 경우의 범칙금으로 이루어져 있음을 감안할 때 전자 부분을 최소화하기 위함이다.) 주어진 양식 내에서 최소 집행비용은 정상비용이고 정상비용은 집행기간이 정상기간일 때 발생하는 비용이므로, 각 활동별 양식은 최소의 정상비용을 지닌 양식으로 선정하고 기간은 선정된 양식의 정상기간으로 한다. 아직 결정되지 않은 프로젝트 비용은 프로젝트 범칙금, $\bar{P} \times z$, 이다. 각 활동별 양식과 기간이 결정되었고 프로젝트 범칙금은 프로젝트 종료시기에 따라 결정되므로 이제 남은 문제는 "단일 양식을 지니며 프로젝트 종료시기 최소화를 목적함수로 하는 RCPSP (a single-mode RCPSP with makespan minimization as its objective)"가 된다. 둘째 단계에서는 상기 문제에 효율적이라 알려진 휴리스틱 기법인 최소 여유시간 규칙(minimum slack rule)을 사용하여 최초의 실행가능일정을 작성한다.

2. 6개의 개선규칙들(improvement rules)

두 번째 단계에서는 첫 단계에서 생성된 실행가능한 일정을 개선시키기 위해 6개의 규칙들을 사용한다. 개선 규칙 1과 2는 A&E 기법의 해당규칙들을 수정 없이 사용할 수 있다. 개선규칙 1은 개별 활동을 보다 적은 집행비용으로 재배정할 수 있는 가를 검토한다. 이 규칙은 재배정되는 개별 활동의 배정요소(시작시기, 양식 및 기간)만을 변경시킬 뿐 다른 활동들의 배정요소는 변경시키지 않는다.

개선규칙 1 : 개별 활동의 일정 변경 (Rescheduling of an activity)

주어진 실행가능한 일정에서 어떤 활동 j가 있다고 하자. 주어진 일정에서 활동 j를 제외한 활동들의 배정요소(시작시기, 양식 및 기간)는 변경시키지 않고 활동 j의 배정요소를 중 한 요소 또는 몇 개의 요소들을 변경시켜 활동 j의 집행비용을 감소시키면서 동시에 모든 제약조건을 충족시키는 일정을 작성할 수 있다면 변경된 일정은 개선된 실행가능한 일정이다.

개선규칙 2는 주어진 일정에서 인접한 활동들의 배정요소를 부분적으로 변경시켜 개선된 실행가능한 일정을 작성하고자 시도한다. 주어진 일정에서 활동 i의 종료시기와 활동 j의 시작시기가 일치한다면 ($t_i = t_j - d_j$), 활동 i와 j는 인접하다고 지칭된다.

개선규칙 2 : 인접한 두 활동의 일정 변경 (Rescheduling of adjacent two activities)

주어진 실행가능한 일정에서 활동 i의 종료시기와 활동 j의 시작시기가 동일한 두 인접활동들이 있다고 하자. 활동 i의 시작시기는 변경하지 않고 활동 i의 집행기간을 1기간 연장하고 활동 j의 시작시기는 1기간 지연하고 집행기간은 1기간 단축시켰다고 하자. 수정된 일정이 프로젝트의 모든 제약조건을 만족시키고 활동 i와 j의 집행비용의 합을 감소시킨다면 수정된 일정은 개선된 실행가능한 일정이다. 역으로 활동 i의 시작시기는 변경하지 않고 활동 i의 집행기간과 활동 j의 시작시기를 1기간씩 단축하고 활동 j의 집행기간은 1기간 연장시켰다고 하자. 수정된 일정이 프로젝트의 모든 제약조건을 만족시키고

활동 i와 j의 집행비용의 합을 감소시킨다면 수정된 일정은 개선된 실행가능한 일정이다.

개선규칙 2의 특이점은 변경 후의 일정에서도 활동 i와 j는 인접한 상태이며 활동 i의 시작시기와 활동 j의 종료시기는 변경되지 않는다는 점이다.

A&E의 개선규칙 3부터 6까지는 참고문헌(4)의 Remark 3.1에 근거하고 있는데 이 Remark는 프로젝트 기간 중 각 자원별로 가용한 자원의 용량이 일정한 경우에만 적용된다. 본고의 수정기법은 프로젝트 기간 중 각 자원의 용량이 시간에 따라 변하는 경우를 다루고 있으므로 규칙 3부터 6까지는 수정을 요한다. 개념정의 3.1은 A&E기법에서 정의된 개념들을 재정리한 것이며 정리 3.2, 3.3 및 3.4는 수정기법의 문제에 맞게 고안된 정리들이다.

개념정의 3.1

R-Demand(p,k): 주어진 일정에 의해 기간 p에 요구되는 자원 k의 양

$$R\text{-Demand}(p,k) = \sum_{j \in SA_p} \sum_{m=1}^{M_j} r_{jm,k} x_{jm}$$

Period-Crash(p): 주어진 일정에서 SA_p에 속한 활동들의 집행기간을 한 기간씩 단축하고 시점 (p+1) 이후에 시작되도록 배정된 활동들의 시작시기를 한 기간씩 단축하는 것.

Period-Uncrash(p): 주어진 일정에서 SA_p에 속한 활동들 중 배정된 집행기간이 배정된 양식의 정상기간이 아닌 활동들의 집행기간을 한 기간씩 연장하고 시점 (p+1) 이후에 시작되도록 배정된 활동들의 시작시기를 한 기간씩 지연하는 것.

SA_p는 주어진 일정에 따라 포함되는 활동들이 변하며 R-Demand(p,k)는 SA_p에 속한 활동들에 의해 요구되는 자원 k의 양을 지칭한다 (문제의 수식화 (5) 참조). Period-Crash(p)는 기간 p에 작업배정된 활동들의 집행기간을 1기간 단축함으로써 프로젝트 일정(fJ)을 1기간 단축하고자 하는 것이며 Period-Uncrash(p)는 작업기간 p에 작업배정된 활동들의 집행기간을 1기간씩 연장함으로써 fJ를 1기간 지연시키는 것이다. 단, Period-Uncrash(p)의 경우 SA_p에 속한 활동들 중 집

행기간이 집행양식의 정상기간인 활동 q가 있다고 하자. 활동 q의 집행기간을 1기간 연장하면 활동 q의 집행비용이 감소하는 것이 아니라 오히려 실행불가능한 일정이 되므로 (문제의 수식화 (7) 참조) SAp에 속한 활동들 중 집행기간이 집행양식의 정상기간인 활동들은 기간을 연장시킬 필요가 없다.

정리 3.2 「실행가능한 Period-Crash(p)」

주어진 실행가능한 일정에서 기간 p, $1 \leq p \leq fJ$,가 존재한다고 하자. 다음의 조건이 충족된다면 Period-Crash(p)를 실행하여 재작성된 일정은 실행가능한 일정이다.

조건 1. SAp에 속한 모든 활동들의 배정된 집행기간은 해당 양식의 단축기간보다 1기간 이상 길다.

조건 2. 각 자원별로 기간 (p+1) 이후의 각 기간별 자원요구량이 선행 기간의 자원가용량을 초과하지 않는다. $R\text{-Demand}(t,k) \leq R_{(t-1)}^k$, $k = 1, \dots, K$, $t = p+1, p+2, \dots, fJ$

조건 1은 자명하며 조건 2는 Period-Crash(p)로 인해 기간 p부터 기간 fJ까지의 자원 요구량이 변화되어도 새로운 일정에서의 각 기간별 자원요구량이 자원의 가용량을 초과하지 않음을 보이고 있다. 나머지 일정에 관한 제약조건은 주어진 일정이 실행가능한 일정이었다면 새로운 자동적으로 충족된다.

정리 3.3 「실행가능한 Period-Uncrash(p)」

주어진 실행가능한 일정에서 기간 p, $1 \leq p \leq fJ$,가 존재한다고 하자. 다음의 조건이 충족된다면 Period-Uncrash(p)를 실행하여 재작성된 일정은 실행가능한 일정이다.

조건 1. 각 자원별로 기간 p 이후의 각 기간별 자원요구량이 후행 기간의 자원가용량을 초과하지 않는다. $R\text{-Demand}(t,k) \leq R_{(t+1)}^k$, $k = 1, \dots, K$, $t = p, p+1, \dots, fJ$

개념정의 3.1에서 정의된 바와 같이 Period-Uncrash(p)의 경우는 SAp에 속한 활동들 중 기간연장이 가능한 활동만을 대상으로 기간연장을 하므로 정리

3.2의 조건 1에 해당되는 조건은 필요없다. 정리 3.3의 조건 1은 정리 3.2의 조건 2와 유사하다.

개선규칙 3은 주어진 실행가능한 일정에서 $z = (fJ - \bar{T}) > 0$ 인 경우에 프로젝트의 한 기간에 실행 중인 활동들의 작업기간을 단축하여 $\bar{P} \times z$ 를 감소시키고자 하는 것이다.

개선규칙 3 : 1기간 단축(One Period Crashing)

주어진 실행가능한 일정에서 $z = (fJ - \bar{T}) > 0$ 인 경우를 가정하자. 주어진 일정에서 「실행가능한 Period-Crash(p)」가 가능하고, Period-Crash(p)로 인한 활동들의 집행비용의 증가분이 프로젝트의 기간당 범칙금(\bar{P})보다 작다면, Period-Crash(p)는 개선된 실행가능한 일정을 작성한다.

개선규칙 4는 주어진 실행가능한 일정에서 $fJ \leq \bar{T} - 1$ 인 경우에 프로젝트의 한 기간에 실행 중인 활동들의 작업기간을 1기간 연장하여 범칙금 지불없이 활동들의 집행비용을 감소시키고자 한다.

개선규칙 4 : 1기간 단축-1 (One Period Uncrashing-1)

주어진 실행가능한 일정에서 $fJ \leq \bar{T} - 1$ 인 경우를 가정하자. 주어진 일정에서 「실행가능한 Period-Uncrash(p)」가 가능하고 Period-Uncrash(p)로 인한 활동들의 집행비용이 감소한다면, Period-Uncrash(p)는 개선된 실행가능한 일정을 작성한다.

개선규칙 5는 주어진 일정에서 $fJ \geq \bar{T}$ 인 경우에 범칙금을 더 지불하고서라도 활동들의 집행비용의 감소를 시도한다.

개선규칙 5 : 1기간 단축-2 (One Period Uncrashing-2)

주어진 실행가능한 일정에서 $fJ \geq \bar{T}$ 인 경우를 가정하자. 주어진 일정에서 「실행가능한 Period-Uncrash(p)」가 가능하고 Period-Uncrash(p)로 인한

활동들의 집행비용 절감이 \bar{P} 보다 크다면 Period-Uncrash(p)는 개선된 실행가능한 일정을 작성한다.

마지막 개선규칙은 주어진 실행가능한 일정에서 Period-Crash와 Period-Uncrash를 동시에 함으로써 개선된 실행가능한 일정을 작성하고자 한다. 개선규칙 6에 필요한 정리는 다음과 같다.

정리 3.4 실행가능한 동시 Period-Crash(p)와 Period-Uncrash(q)

주어진 실행가능한 일정에서 두 기간, p와 q를 가정하자. 만약 다음의 조건들이 충족된다면 Period-Crash(p)와 Period-Uncrash(q)를 동시에 적용하여 얻어지는 새로운 일정은 실행가능한 일정이다.

p < q인 경우

조건 1. SAp에 속한 각 활동의 배정된 집행기간은 해당 양식의 단축기간보다 1기간 이상 길다.

조건 2. 각 자원별로 기간 p+1부터 q까지의 각 기간별 자원요구량이 선행 기간의 자원 가용량을 초과하지 않는다. $R\text{-Demand}(t,k) \leq R_{(t-1)}^k, k = 1, \dots, K, t = p, p+1, \dots, q$

p > q인 경우

조건 1. p < q인 경우의 조건 1과 동일

조건 2. 각 자원별로 기간 q부터 p-1까지의 각 기간별 자원요구량이 후행 기간의 가용량을 초과하지 않는다. $R\text{-Demand}(t,k) \leq R_{(t+1)}^k, k = 1, \dots, K, t = q, q+1, \dots, p-1$

정리 3.4는 (i) p < q의 경우에는 Period-Crash(p)를 실시하고 새로 작성된 일정에서 Period-Uncrash(q-1)을 실행하는 것과 동일하며 (ii) p > q인 경우에는 Period-Crash(p)를 실시하고 새로이 작성된 일정에서 Period-Uncrash(q)를 실행하는 것과 동일하다. 정리 3.4에서 조건 2는 자원제약조건 (문제의 수식화 (5) 참조)에 관한 것으로서 각 기간별 자원 요구량의 변화는 (i) p < q인 경우에 기간 p부터 q-1까지 (ii) p > q인 경우는 q+1부터 p까지에 한정하여 발생한다. 그리고 Period-Crash(p)와 Period-Uncrash(q)의 동시 적용은 fJ를 변경시키지 않는다.

개선규칙 6은 주어진 실행가능한 일정에 「실행가능한 동시 Period-Crash(p)와 Period-Uncrash(q)」를 실행하여 활동들의 집행비용의 절감을 시도한다.

개선규칙 6: Period-Crash와 Period-Uncrash의 동시 적용 (Simultaneous one period crashing and uncrashing)

주어진 실행가능일정에서 정리 3.4의 조건들이 성립하고 Period-Crash(p)로 인한 활동들의 집행비용 증가가 Period-Uncrash(q)로 인한 활동들의 집행비용 감소보다 작다면 Period-Crash(p)와 Period-Uncrash(q)의 동시 적용은 개선된 실행가능한 일정을 작성한다.

3. 6개의 개선규칙 반복 적용 후의 새로운 실행 가능한 일정 작성

이 부분은 A&E기법과 동일하므로 약술하고자 한다. 두 번째 단계가 종료하면 기법은 첫 번째 단계로 되돌아가 새로운 일정을 작성한다. 만약 n번째 경로의 첫 번째 단계에서 작성된 일정이 (n+1)번째의 첫 번째 단계에서 작성된 일정과 동일하다면 (n+1)번째 경로는 n번째 경로와 동일하게 된다. 이를 피하고자 다음과 같은 조치를 취한다. (i) 첫 번째 경로 첫 번째 단계에서 최초의 실행 가능한 일정을 작성할 때 2차원의 M_INDEX란 행렬을 다음과 같이 작성한다: 만약 활동 j가 양식 mj로 배정되었으면 M_INDEX(j)[mj]에 '최초 일정의 프로젝트 총비용'의 값을, 그렇지 않은 경우에는 '충분히 큰 값'을 기입한다. (ii) M_INDEX 값은 현 경로에서 일정이 개선될 때마다 개선된 일정에서 활동 j가 양식 mj로 배정되었으면 M_INDEX(j)[mj]에 '개선된 일정의 프로젝트 총비용'의 값으로 개정하고, 그렇지 않은 경우에는 수정을 하지 않는다. 그러므로 해당 경로의 끝에 M_INDEX의 값이 크다는 것은 최근에 해당 활동이 해당 양식으로는 사용되지 않았다는 것을 의미하며 M_INDEX 값이 작다는 것은 그 반대의 경우를 의미한다. (iii) 현 경로에서 다음 경로의 최초 실행가능일정을 작성할 때는 M_INDEX 중 가장 큰 값을 미리 설정한 숫자(본고에서는 (J-2)×0.1로 설정하였다.)만큼 선택한다. (단, 동일 활동에서는 가장 큰 값을 하나만 선택한다.) 예를 들어 M_INDEX(h)[mh]가 선택되었다고 하자. 이 경우 활동 h의 양식은 mh로 변경하고, 활동 h의 기간은 양식

mh의 정상기간과 단축기간의 사이에서 무작위로 설정한다. M INDEX의 값이 선택되지 않은 활동들의 양식과 기간은 변경하지 않는다. 이처럼 모든 활동들의 양식과 기간이 설정되면 최초 경로에서와 마찬가지로 '최소 여유 시간 규칙'을 적용하여 해당 경로의 최초 실행가능한 일정을 작성한 후 두 번째 단계로 진입한다. 이러한 과정은 미리 설정된 컴퓨터 실행시간이 경과할 때까지 반복한다.

IV. 모의계산

수정 휴리스틱 기법의 효율성 검증은 최적기법 및 동일 문제에 대한 다른 휴리스틱 기법들과의 비교를 통해 이루어질 수 있다. 본 문제에 관한 최적해법은 안태호 [1]을 사용하였고, 본 문제에 대한 휴리스틱 기법은 본고의 수정 휴리스틱 기법이 최초인 관계로, 안태호[1]의 최적해법을 변형하여 두 개의 새로운 휴리스틱 기법을 만들어 비교하였다. 안태호[1]의 최적해법을 변형한 첫 번째 휴리스틱 기법은 최적해법에 시간제약을 가하여 시간 내 발견된 가장 좋은 일정을 답으로 제시하는 방법이다. 안태호[1]의 최적해법은 "부분가능일정 (partial feasible schedule)"란 개념에 기초하고 있는 데, "부분가능일정"의 개념을 사용하고 있는 최적해법들은 최적해를 탐색하는 데 소요되는 시간을 줄이기 위해 다양한 바운딩 규칙 (bounding rule)들을 사용한다. 이런 부류의 최적해법을 휴리스틱 기법으로 변형하였을 때의 단점은 주어진 시간 내에 탐색된 모든 부분일정이 바운딩 규칙에 의해 제거되어, 휴리스틱 기법이 종료될 때까지 실행가능한 일정을 하나도 찾지 못할 수 있다는 점이다. 이러한 단점을 없애기 위해 최적해법에서 변형한 두 번째 휴리스틱 기법에서는 안태호[1]에서 사용된 모든 바운딩 규칙들을 제거하였다. 표기의 편의상 본고의 수정해법은 AH (Ahn's heuristic procedure)로, 안태호[1]의 최적해법은 AE (Ahn's exact solution procedure)로, AE에서 변형한 첫 번째 휴리스틱 기법은 TAE1 (Truncated version

of Ahn's Exact Solution Procedure)로, 그리고 마지막 휴리스틱 기법은 TAE2로 칭하기로 한다.

1. 모의일정작성

모의일정은 기본적으로 참고문헌[4]에 적용한 자료를 본고의 문제에 적합하도록 수정하였다. 참고문헌[4]에 사용된 자료는 10개씩의 일정으로 이루어진 10개의 일정군으로 구성되어 있으며, 프로젝트의 비가상 활동의 수, J-2,는 10에서 50에 이른다. 모든 일정의 공통점을 약술하면, $M_j = 2, K = 2, ND_{jm}, CD_{jm} = 2, C$ (Complexity of Network) = 1.8이다. 본고에서는 참고문헌[4]에서와 같은 방식으로 (J-2)가 75인 10개의 일정들을 작성하고 추가하여 본고에서 사용된 문제는 총 11개의 일정군의 110개의 일정으로 이루어져 있다. 참고문헌[4]의 일정에서는 자원의 가용량이 프로젝트 기간 중 변하지 않는 경우이었으므로 본고에서는 다음과 같이 수정하였다.

표 1. 프로젝트 기간별 각 자원의 가용량 변경
Table 1. modification on the per period availability of each resource type

R^k	참고문헌(4)의 일정에서 자원 k의 가용량
R_p^k	본고의 문제에서 자원 k의 기간 p에서의 가용량 k가 홀수인 경우 $R_p^k = R^k + 1, \text{ if } 11 \leq p \leq 20,$ $51 \leq p \leq 60, 91 \leq p \leq 100, \dots$ $R_p^k = R^k - 1, \text{ if } 31 \leq p \leq 40,$ $71 \leq p \leq 80, 111 \leq p \leq 120, \dots$ $R_p^k = R^k, \text{ 기타의 경우}$ k가 짝수인 경우 $R_p^k = R^k + 1, \text{ if } 21 \leq p \leq 30, 61 \leq p \leq 70,$ $101 \leq p \leq 110, \dots$ $R_p^k = R^k - 1, \text{ if } 41 \leq p \leq 50, 81 \leq p \leq 90,$ $121 \leq p \leq 130, \dots$ $R_p^k = R^k, \text{ 기타의 경우}$

2. 모의계산결과와 보고

AH, AE, TAE1 및 TAE2는 C 언어로 코드화하였다. 계산에는 Intel pentium 200Mhz CPU를 지닌 컴퓨터를 사용하였다. 각 휴리스틱 기법에 주어진 시간

1) "부분가능일정"과 "바운딩규칙"에 대한 개념은 Sprecher [12]에서 자세히 설명되어 있다.

제약은 $(J-10)/2$ 초이다. 각 휴리스틱 기법의 효율성은 오차백분율을 사용하여 측정하였는데 오차백분율은 "100 × [휴리스틱 기법에 의한 일정의 총비용-기준 일정의 총비용]/기준 일정의 총비용"으로 정의하였다. (J-2)가 20 이하인 일정군들은 AE를 적용하여 최적 일정을 찾아 그 일정들을 기준 일정으로 삼았으며, (J-2)가 20을 초과하는 경우에는 AE를 적용하여 최적일정을 찾는 데 소요되는 시간이 너무 큰 관계로 각 휴리스틱 기법에 의한 일정 중 일정의 총 비용이 가장 저렴한 일정을 기준일정으로 택하였다. 본 모의계산에서는 참고문헌(4)의 실험에서 사용된 \bar{P} 값을 수정없이 사용하여 실험하여보고 그리고 10배로 증가시켜 보기도 하였다. 이는 프로젝트 지연은 프로젝트 수행기관의 신뢰도와 깊은 관련이 있기 때문에 큰 값의 \bar{P} 에서 각 기법의 효율을 측정하기 위함이다. 표 2와 3에서 첫 번째 열의 백분율은 찾아낸 10개 일정의 평균 오차백분율을, 그리고 TAE1의 경우 ()속의 숫자는 주어진 시간 내에 실행가능한 일정을 하나도 작성치 못한 경우를 나타낸다. (이 경우 실행가능한 일정만을 가지고 평균 오차백분율을 구하였다.) 두 번째 열의 숫자는 해당 휴리스틱 기법이 찾아낸 일정 중 총비용이 기준 일정의 총비용과 동일한 일정의 수를 나타낸다. 그러므로, 표2에서는 최적일정의 수를, 표3에서는 3개의 휴리스틱 기법이 찾아낸 일정 중 가장 좋은 일정의 수를 의미한다.

표 2. 휴리스틱 기법들의 효율비교, (J-2) ≤ 20
Table 2. Comparison of Heuristic Procedures' Efficiencies, (J-2) ≤ 20

J-2	\bar{P} 를 수정 않은 경우			\bar{P} 를 10배로 수정		
	AH	TAE1	TAE2	AH	TAE1	TAE2
10	0.21	0.58	22.95	0.22	15.37	127.77
	8	8	0	8	8	1
12	0.60	1.87	17.91	0.73	2.26	106.27
	4	3	0	3	3	0
14	0.48	5.69	23.87	0.14	13.53	128.30
	4	2	0	8	2	0
16	0.65	9.58	36.26	1.01	49.95	284.53
	5	1	0	4	1	0
18	0.46	9.05(2)	44.09	0.40	9.87 (3)	304.34
	3	0	0	3	0	0
20	0.50	12.21(3)	32.45	0.78	39.23(4)	228.97
	4	0	0	1	0	0

표 3. 휴리스틱 기법들의 효율비교, (J-2) ≥ 25
Table 3. Comparison of Heuristic Procedures' Efficiencies, (J-2) ≥ 25

J-2	\bar{P} 를 수정 않은 경우			\bar{P} 를 10배로 수정		
	AH	TAE1	TAE2	AH	TAE1	TAE2
25	0.00	37.12(9)	33.89	0.00	327.38(9)	256.76
	10	0	0	10	0	0
30	0.00	N/A(10)	31.93	0.00	N/A(10)	250.95
	10	0	0	10	0	0
40	0.00	N/A(10)	28.76	0.00	N/A(10)	202.94
	10	0	0	10	0	0
50	0.00	N/A(10)	26.38	0.00	N/A(10)	190.88
	10	0	0	10	0	0
75	0.00	N/A(10)	23.60	0.00	N/A(10)	188.97
	10	0	0	10	0	0

표 2와 3에 나타나 있듯 본고에서 제시한 휴리스틱 기법은 비교대상이 된 기법들 중 가장 효율적이다.

V. 결론

본고에서는 학계에 최근 보고된 RCPSPMCM 문제 중 자원의 가용량이 프로젝트 기간 중 일정치 않은 경우의 휴리스틱 기법을 소개하였다. 이 문제의 최적해법은 안태호(1)에 의해 개발되었으나, 휴리스틱 기법은 아직 학계에 보고된 바 없다. 본고에서 소개된 휴리스틱 기법은 RCPSPMCM 문제 중 자원의 가용량이 프로젝트 기간 중 일정한 경우를 다룬 Ahn and Erenguc(4)의 휴리스틱 기법의 기본 틀을 사용하여 세부적인 규칙들을 본 문제에 합당하도록 개선한 것이다. 본 휴리스틱의 효율성은 (i) 문제의 크기가 작은 경우에는 안태호(1)의 최적해법에 의한 최적일정과의 비교를 통해, (ii) 문제의 크기가 큰 경우에는 안태호(1)의 최적해법을 변형하여 만든 휴리스틱 기법들과의 상대적 비교를 통하여 검증하였다. 총 110개의 가상 일정을 작성하여 검토한 바, 본고의 휴리스틱 기법의 효율성이 높은 것으로 나타났다.

참고문헌

- [1] 안태호, "자원의 양이 제약되고 기간단축이 가능한 복수의 양식을 지닌 단일 프로젝트 일정문제 - 자원의 용량이 시간에 따라 변하는 경우의 최적해법," *승실경영경제연구*, 제 25집, pp. 103-119, 1995
- [2] Ahn, T., "Resource Constrained Project Scheduling Problem with Time/Cost Trade-Offs: the Multimode Case", PhD Dissertation, University of Florida, 1994
- [3] Ahn, T. and Erenguc, S. S., "The Resource Constrained Project Scheduling Problem with Multiple Crashable Modes: An Exact Solution Method", Working Paper Series, #95-101, Department of Decision and Information Sciences, University of Florida, 1995
- [4] Ahn, T. and Erenguc, S. S., "The Resource Constrained Project Scheduling Problem with Multiple Crashable Modes: A Heuristic Procedure," *European Journal of Operational Research*, Vol. 107, No. 2), pp. 250-259, 1998
- [5] Demeulemeester, E.L. and Herroelen, W.S., "A Branch and Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem," *Management Science*, Vol. 38, pp. 1803-1818, 1992
- [6] Patterson, J.H. and Huber, W.D., "A Horizon-Varying, Zero-One Approach to Project Scheduling," *Management Science*, Vol. 20, pp. 990-998, 1974
- [7] Patterson, J.H. and Roth, G., "Scheduling a Project under Multiple Resource Constraints: A Zero-One Approach," *AIIE Transactions*, Vol. 8, pp. 449-456, 1976
- [8] Patterson, J.H., Slowinski, R., Talbot, F.B. and Weglarz, J., "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, pp.3-28, 1989
- [9] Patterson, J.H., Slowinski, R., Talbot, F.B. and Weglarz, J., "Computational Experience with a Backtracking Algorithm for Solving a General Class of Precedence and Resource Constrained Scheduling Problems," *European Journal of Operational Research*, Vol. 49, No. 1, pp. 68-79, 1990
- [10] Stinson, J.P., Davis, E.W. and Khumawala, B.M., "Multiple Resource-Constrained Scheduling Using Branch and Bound," *AIIE Transactions*, Vol. 10, pp. 252-259, 1978
- [11] Talbot, F.B., "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case," *Management Science*, Vol. 28, pp. 1197-1210, 1982
- [12] Sprecher, A., "Resource-Constrained Project Scheduling: Exact Methods for the Multi-Mode Case," *Lecture Notes in Economics and Mathematical Systems*, 409, Springer-Verlag, Berlin, 1994

저자소개

안태호
 1984 : 고려대학교 독문과 졸업(학사/석사)
 1991 : 미 펜실베니아 주립대학 경영대학원 MBA
 현재 : 승실대학교 경영학부 조교수
 관심분야 : 일정관리, 시스템과학, 통합자원관리