

R&D Project Portfolio 선정 문제

†안태호* · 김명관**

R&D Project Portfolio Selection Problem

†Taeho Ahn* · Myung Gwan Kim**

■ Abstract ■

This paper investigates the R&D project portfolio selection problem. Despite its importance and impact on real world projects, there exist few practical techniques that help construct a non-dominated portfolio for a decision makers satisfaction. One of the difficulties constructing the portfolio is that such project portfolio problem is, in nature, a multi-attribute decision-making problem, which is an NP-hard class problem. This paper investigates the R&D project portfolio selection problem. Despite its importance and impact on real world projects, there exist few practical techniques that help construct a non-dominated portfolio for a decision makers satisfaction. One of the difficulties constructing the portfolio is that such project portfolio problem is, in nature, a multi-attribute decision-making problem, which is an NP-hard class problem.

In order to obtain the non-dominated portfolio that a decision maker or a user is satisfied with, we devise a user-interface algorithm, in that the user provides the maximum/minimum input values for each project attribute. Then the system searches the non-dominated portfolio that satisfies all the given constraints if such a portfolio exists. The process that the user adjusts the maximum/minimum values on the basis of the portfolio found continues repeatedly until the user is optimally satisfied with. We illustrate the algorithm proposed, and the computational results show the efficacy of our procedure.

Keywords : Project Portfolio

논문접수일 : 2007년 07월 02일 논문게재확정일 : 2007년 10월 15일

* 숭실대학교 경영학부

** 대전시전략산업기획단

† 교신저자

1. 서 론

과학기술의 본질이 수적 개념에서 기술 중심의 개념으로 변화된 오늘날, 연구과제관리의 합리적 관리는 연구개발 능력의 핵심으로 볼 수 있다. R&D의 합리적 관리는 “어떤 기술들을 어떻게 확보하여야 하는가”라는 이슈를 지니고 있다. 이 이슈에 관한 접근법은 크게 두 가지로 볼 수 있다. 첫 번째 접근방식은 개별 프로젝트 또는 사업의 평가로서 타당성 분석 등의 기법이 활용되고 있으며, 두 번째 접근방식은 일정한 기간동안 수행하여야 할 프로젝트군(Project Portfolio)을 선정하는 것이다. 본 연구는 후자에 속하는 것으로, 한정된 예산 등을 고려하여 복수의 프로젝트들 중에 실행할 프로젝트들을 선정하는 방법에 관한 것이다. 이러한 방법론의 특징은 선정된 개별 프로젝트 자체보다는 선정된 프로젝트군의 속성에 초점을 맞춘다는 점이다. 포트폴리오(Portfolio)의 개념을 주식에 적용하여 설명하여 보자. 고수익이 예상되지만 주식시장에 매우 민감한 주식만을 지니고 있다면 IMF와 같은 상황에서는 매우 위험할 수 있다. 그러므로 개별 주식들의 수익을 및 리스크 등을 고려하여 나에게 적합한 주식 포트폴리오를 구성하는 것이 바람직한 것이다. 본 논문에서는 연구과제 관리에 포트폴리오의 개념을 접목시켜 최적 프로젝트 포트폴리오를 구성하는 기법을 소개하고자 한다.

포트폴리오 개념의 프로젝트 선정모형은 다른 모형에 비해 어떤 장단점을 가지고 있는 것일까? Souder[15]는 연구개발프로젝트의 선정모형을 선별(Screening)모형, 우선순위 결정을 위한 평가(Evaluation)모형, 이들 프로젝트를 전체로 묶어 최적화 구성을 찾아내기 위한 포트폴리오(Portfolio) 분석 모형으로 구분하고 있다. 선별모형은 프로젝트 선정에서 고려해야 할 요소들을 파악하고 각 프로젝트를 이 요인에 의해 평가하여 선별하는 것으로 사용이 편리하고 자료 요구량이 많지 않고 쉽게 프로젝트를 선별할 수 있다는 점에서 장점이 있으나 평가가 주관적이며 평가자에 의해 평가의 차이가 크다

는 단점이 있다고 Souder는 정리하였다. 평가모형은 평가자료의 요구량도 많고 정확성도 높고 요구하고 있으나 대신 프로젝트 특성을 심도 있게 분석하는 것이 가능하다. 이 모형은 적용이 간단하고 재무 자료를 이용하므로 실용적이라는 장점이 있으나 단일 지표로는 성공확률, 수익, 비용간 대체관계를 설명할 수 없으며 여러 개의 속성을 동시에 고려할 수 없다는 단점이 있다. 포트폴리오 분석모형은 개별 프로젝트의 선정여부가 프로젝트 별로 독립적으로 이루어지는 것이 아니라 포트폴리오 입장에서 선정되는 것으로 연구 프로젝트의 특성인 불확실성, 연속적 의사결정, 자원제약 등 많은 요소를 동시에 해결할 수 있다고 Souder는 정리하였다.

일반적으로 포트폴리오 분석모형에 의한 프로젝트 선정은 객관적인 기준에 의해 이루어지므로 특정 프로젝트에 대한 개인적인 선호도에 영향을 받지 않는다는 점과 조직의 전략적 방향에 맞게 포트폴리오를 조정하여 자원의 최적배분이 가능하다. 아울러 의사결정자의 위험에 대한 선호도를 반영하여 리스크 수준별로 포트폴리오를 구성할 수 있으며 프로젝트별 수행기간이 다른 경우 기간별로 자원의 최적 활용이 가능한 포트폴리오를 도출할 수 있다. 또한 많은 연구프로젝트가 프로젝트들 사이에 상호 의존성(Interdependent)을 갖고 있는 점을 감안하여 이러한 상호의존성을 연구 성과나 자원측면에서 반영한 최적의 포트폴리오를 구성할 수 있다. 그러나 이 모형의 적용에는 상당한 전문적 지식과 활용에 따른 제반 문제를 잘 인식하여 대처하는 것이 필요하고 많은 자료의 확보가 선행되어야 하며 정확하고 객관적인 자료가 없으면 모형의 적용가치가 없어지는 단점이 있다. 이 모형에 관련된 세부 기법들이 많이 개발되어 있으나 [2] 복잡성, 자료 요구량 등의 제약, 의사결정자 판단이나 경험 및 직관 등을 반영하기 어렵기 때문에 별로 활용되지 않고 있다[10] 최근 컴퓨터 성능 발전과 관련 분야의 연구 활성화로 확산되고 있다[6].

2. 기존연구의 고찰

R&D 포트폴리오 선정에 관한 수리모형은 점수 평점법[3], 단일목적 및 다목적계획법 등을 중심으로 진행되어 왔다[12, 18] 최근의 포트폴리오 선정 모형과 관련한 주요 문제는 비열등 포트폴리오를 탐색하는 연산 속도와 프로젝트간의 상호 의존성 고려 여부 및 프로젝트 착수시간(Start Time)이란 변수의 고려 여부 등이다. Masood et al.[9]은 0-1 Goal Programming Model을 활용하여 28개의 후보 프로젝트를 대상으로 비열등 포트폴리오를 구성하여 보았는데, 이때 문제를 단순화하기 위해 상호 의존성은 고려하지 않았다. Ringuest et al.[7] 현재의 포트폴리오에서 새로운 프로젝트를 추가할 것인지 여부에 대한 분석을 실시한 바 있다. 여기서 그들은 모형을 개발하기 위해 먼저 5개의 프로젝트를 갖고 32개의 포트폴리오를 만들어 시도했으며 모형이 개발된 후에 24개 프로젝트(약 1,700만개의 포트폴리오 검색)와 30개 프로젝트(약 10억개의 포트폴리오 검색)를 시도하였다. 아울러 그들은 이러한 문제를 해결하기 위해서는 휴리스틱에 의한 접근방법과 연계하여 사용하는 것이 바람직하다고 언급하고 있다. Stummer et al.[17] 사용자 인터페이스를 골격으로 하는 방법론을 개발하였다. 그들의 방법은 전체 열거법을 사용하여 모든 비열등 프로젝트 포트폴리오를 구한 후, 사용자가 그래픽을 통하여 자신이 원하는 포트폴리오를 택하는 것이다. 그들은 20개의 프로젝트들을 대상으로 방법론을 적용한 결과, 모든 비열등 포트폴리오를 탐색하는 데 펜티엄 233 PC로 1분 이내의 시간이 소요되었다고 보고하였다.

3. 수리적 모형개발 및 기본 알고리즘

본 연구에서는 J개의 후보 프로젝트들로부터 실행할 프로젝트들의 포트폴리오를 구성하는 문제를 다루고 있다. 각 프로젝트들은 P개의 긍정적인 속

성(예 : 경제성, 기술력 축적)들과 N개의 부정적인 속성(예 : 예산, 리스크)들을 지니고 있으며, 의사결정자의 관심은 어떻게 포트폴리오를 구성하여야 포트폴리오 자체의 긍정적인 속성과 부정적인 속성을 통제할 수 있는 가로 규정한다. 포트폴리오의 i번째 속성값은 선정된 프로젝트들의 i번째 속성값들의 합이라 가정한다. 본 모형에서는 어떤 프로젝트가 선정되면 어떤 프로젝트는 선정될 수 없다는 상호 배타적 프로젝트 집합도 고려하였다.

본 모형의 주요 가정을 정리하면 다음과 같다.

- 1 : 개별 프로젝트들은 다수의 속성을 지닌다.
- 2 : 프로젝트의 속성값들은 사전에 알려져 있으며 확정적이다.
- 3 : 포트폴리오의 i번째 속성값은 선정된 프로젝트들의 i번째 속성값들의 합이라 가정한다.

3.1 수리 모형

$$\max \sum_{j=1}^J p a_j^1 \times x_j \quad (1)$$

$$\max \sum_{j=1}^J p a_j^2 \times x_j \quad (2)$$

.....

$$\max \sum_{j=1}^J p a_j^P \times x_j \quad (3)$$

$$\min \sum_{j=1}^J n a_j^1 \times x_j \quad (4)$$

$$\min \sum_{j=1}^J n a_j^2 \times x_j \quad (5)$$

.....

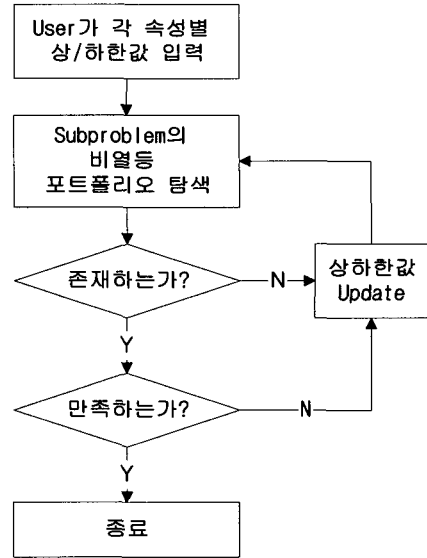
$$\min \sum_{j=1}^J n a_j^N \times x_j \quad (6)$$

$$\text{st } x_j = \begin{cases} 1, & \text{if project } j \text{ is chosen.} \\ 0, & \text{o.w.} \end{cases}, \quad j = 1, \dots, J \quad (7)$$

$$\sum_{j \in M E_k} x_j = N_k, \quad k = 1, \dots, K \quad (8)$$

J는 프로젝트의 수를 나타내는 상수이고, j는 프로젝트의 인덱스로서 1부터 J까지의 값을 지닌다. $p a_j^k$ 는 프로젝트 j의 k번째 긍정적인 속성치,

na_k^j 는 프로젝트 j의 k번째 부정적인 속성치를 나타낸다. 본 수식에서는 프로젝트들이 P개의 긍정적인 속성과 N개의 부정적인 속성을 가지고 있다고 가정한다. 수식의 변수는 x_j 로서, 프로젝트 포트폴리오에 선정되면 1의 값을, 아니면 0의 값을 지닌다. 목적식 (1)부터 목적식 (6)까지는 수식의 목적이 프로젝트 포트폴리오에서 긍정적인 속성의 합을 최대화하고, 부정적인 속성의 합의 최소화함을 보여주고 있으며, 제약조건 (7)은 변수 x_j 가 0-1변수임을 나타내고 있다. 상호 배타적인 프로젝트들에 관한 제약식은 식 (8)에 정리되어 있다. k는 상호 배타적인 프로젝트 집합에 관한 인덱스이며, ME_k 는 k번째 상호 배타적인 프로젝트 집합을, N_k 는 k번째 상호 배타적인 프로젝트 집합에서 선정되어야 하는 프로젝트의 수를 나타낸다.



<그림 1> 알고리즘의 흐름도

3.2 기본 알고리즘 및 하위 문제(Subproblem)

본 모형은 전형적인 다목적 의사결정모형으로서, 단일 최적가능해(Optimal Feasible Solution)는 존재하지 않으며, 다수의 비열등해(Non-dominated Solution)들이 존재한다. 의사결정자는 다수의 비열등해 중에서 자신이 원하는 비열등해를 찾아야 하는데, 문제는 비열등해의 수가 경우에 따라서는 매우 많아 이들을 하나씩 검토한다는 것이 원천적으로 불가능할 수 있다. 이러한 문제를 해결하기 위해 본 연구에서는 사용자 인터페이스(User Interface)를 도입한, 다단계 접근방식을 채택하였다. 즉, 의사결정자는 포트폴리오의 각 속성에 관한 자신의 목표치를 입력하고, 프로그램은 이를 분석하여 만족시키는 포트폴리오가 존재하면 이들 중 비열등 포트폴리오를 제시하고, 존재하지 않으면 가능한 포트폴리오가 없음을 보고한다. 의사결정자는 자신의 목표치를 반복적으로 수정함으로써 중국에는 원하는 비열등 프로젝트 포트폴리오를 찾게 된다. 이를 흐름도로 정리하면 <그림 1>과 같다.

의사결정자(User)에 의해 긍정적인 속성들에는 하한값이, 부정적인 속성들에게는 상한값이 주어지

게 되며, 프로그램에서는 이들 상/하한값을 모형에 첨가하여 비열등해를 탐색하게 된다. 상/하한값이 첨가된 문제를 하위문제(Subproblem)이라 정의하고 이를 수리 모형으로 표현하면 다음과 같다.

$$\begin{aligned} \max/\min & \quad (1), (2), \dots, (6) \\ \text{st} & \quad (7), (8) \end{aligned}$$

$$\sum_{j=1}^J pa_j^1 \times x_j \geq LB(pa^1) \quad (9)$$

$$\sum_{j=1}^J pa_j^2 \times x_j \geq LB(pa^2) \quad (10)$$

.....

$$\sum_{j=1}^J pa_j^p \times x_j \geq LB(pa^p) \quad (11)$$

$$\sum_{j=1}^J na_j^1 \times x_j \leq UB(na^1) \quad (12)$$

$$\sum_{j=1}^J na_j^2 \times x_j \leq UB(na^2) \quad (13)$$

.....

$$\sum_{j=1}^J na_j^N \times x_j \leq UB(na^N) \quad (14)$$

하위문제에서 $LB(pa^p)$ 는 p번째 긍정적 속성의

하한값을, $UB(na^i)$ 은 n 번째 부정적 속성의 상한값을 나타낸다.

기본 알고리즘에서 나타나 있듯이 본 연구의 핵심은 하위문제에서 비열등 포트폴리오를 신속히 찾아내는데 있다.

3.3 하위 문제(Subproblem)의 특징

하위문제는 전형적인 NP(Nonpolynomial) Hard 계열의 문제로서, 후보 프로젝트가 10개라면 총 가능한 포트폴리오의 수는 $2^{10} = 1,024$, 20개 라면 약 100만, 30이라면 약 10억 개에 이른다. 하위문제는 NP Hard에 속하므로, 후보 프로젝트의 수가 일정 수준 이상이면 현실적인 시간 내에 비열등 포트폴리오를 찾을 수 있는 방법은 없다. 이러한 NP Hard 계열의 문제를 접근하는 방식은 첫째, 후보 프로젝트의 수가 적은 경우에는 최적화 기법을 사용하고, 둘째 후보 프로젝트의 수가 많은 경우에는 휴리스틱 기법을 사용하는 것이다. 본 연구에서는 하위문제의 비열등 포트폴리오를 찾는 최적화 기법을 논하는데, 이에 대한 방법으로는 선형계획법(Linear Programming)을 응용하는 방법과 전체 열거법(Total Enumeration Technique)을 사용하는 방법 두 가지를 고려할 수 있다. 선형계획법을 응용하는 방법은 후보 프로젝트의 수가 30개만 초과되어도 0-1 정수변수가 30개이므로, 현실적으로 불가하다. 그러므로 본 연구에서는 전체 열거법을 사용한다.

4. 하위문제의 알고리즘

전체 열거법이란 가능한 모든 해를 체계적으로 나열하여 탐색하는 기법으로, 본 연구에서는 모든 프로젝트가 채택되지 않는 조합에서 시작하여 모든 프로젝트가 채택되는 조합까지를 탐색한다. 프로젝트의 수가 총 3개인 경우를 예로 들어 정리하면 <표 1>과 같다(<표 1>에서 0은 해당 프로젝트가 채택되지 않는 경우를, 1은 채택됨을 의미한다).

<표 1> 전체 열거법의 예

No.	프로젝트1	프로젝트2	프로젝트3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

탐색하여야 하는 조합의 수는 프로젝트 수의 제곱이 되며, 본 연구에서 사용한 기본 알고리즘은 다음과 같다.

4.1 하위문제의 기본 알고리즘

- 단계 0 : 모든 프로젝트가 채택되지 않는 조합부터 검색한다. 발견된 실행 가능한 해(Feasible Solution)의 집합은 $\{ \}$ 으로 정의한다.
- 단계 2 : 해당 조합이 하위문제의 모든 제약식을 충족시키는 가를 조사한다. 만약 모든 제약식을 충족시키면 단계 3으로, 그렇지 않으면 단계 4로 간다.
- 단계 3 : 실행 가능한 해의 집합이 공집합이면, 검토 중인 조합을 실행 가능한 해에 포함시키고, 하위 문제에서 각 속성의 합에 대한 제약 ($LB(pa^1), \dots, LB(pa^p), UB(na^1), UB(na^v)$)의 값을 변경하되, $LB(pa^1)$ 의 경우에는 해당 조합에서 선택된 프로젝트들의 첫 번째 긍정적 속성값의 합으로 하고, 나머지 것들도 이에 준하여 변경한다. 단계 5로 간다.
- 단계 4 : 현재 검토 중인 조합의 목적함수값이 실행 가능한 해의 집합에 있는 조합에 비하여 하나라도 개선되었으면 단계 3에서와 같이 현재 검토 중인 조합을 실행 가능한 해의 집합에 포함시키고 기존의 조합을

삭제하며, 단계 3에서와 같이 제약식을 변경하고 단계 5로 간다.

단계 5: 현재 검토 중인 조합이 마지막 조합이 아니라면 다음 조합을 택한 후 단계 2로 간다. 마지막 단계라면 종료한다.

위의 알고리즘은 전체 조합에서 하나의 비열등 포트폴리오를 찾기 위한 것으로, 제약조건을 충족시키는 조합이 하나라도 존재하면 반드시 비열등 조합을 찾는다.

본고에서는 몇 개의 개선규칙(Bounding Rule)들을 도입하여 기본 알고리즘의 연산 부담을 감소시켰다.

4.2 개선규칙 1

첫 번째 개선규칙은 프로젝트의 개별 속성을 활용한 규칙이다. 내용을 정리하면 아래와 같다.

프로젝트의 수가 J인 임의의 조합이 있다고 가정하자.

- (a) 현재의 조합에서 k ($k < J$) 번째까지 선정/비선정된 프로젝트 중 선정된 프로젝트의 i번째 부정적 속성들의 합이 하위문제에 입력된 i번째 부정적 속성의 상한값, $UB(na^i)$ 을 초과한다면, 현재의 조합은 제약조건을 충족시킬 수 없다.
- (b) 현재의 조합에서 k ($k < J$) 번째까지 선정/비선정된 프로젝트 중 선정된 프로젝트의 m번째 긍정적 속성들의 합을 A라 하자. 그리고 (k+1)부터 J까지의 프로젝트들이 모두 선정될 경우 그들의 m번째 긍정적 속성들의 합을 B라 하자. 만약 (A+B)의 값이 하위문제에 입력된 m번째 긍정적 속성의 하한값, $LB(pa^m)$ 에 미달한다면, 현재의 조합은 제약조건을 충족시킬 수 없다.

개선규칙 1은 자명하다. 1부터 k까지의 부분조합에 변화없이 (k+1)부터 J까지의 부분조합에 변화

를 주어도 모두 개선규칙 1이 동일하게 적용되므로, 총 $(J-k)2^k$ 의 조합은 검색할 필요가 없다.1)

4.3 개선규칙 2

두 번째 개선규칙은 긍정적 속성과 부정적 속성의 조합을 활용한 규칙이다.

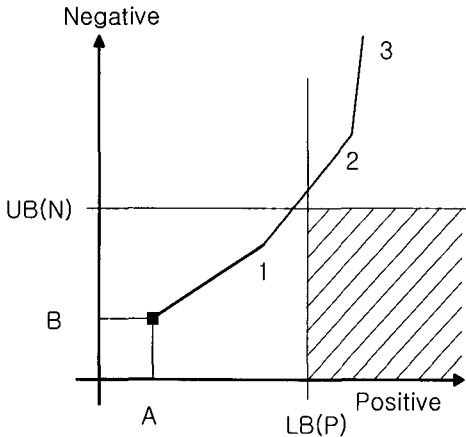
프로젝트의 수가 J인 임의의 조합이 있다고 가정하자. 현재의 조합에서 k ($k < J$) 번째까지 선정/비선정된 프로젝트 중 선정된 프로젝트의 i번째 긍정적 속성의 합을 A라 하고, 선정된 프로젝트의 j번째 부정적 속성의 합을 B라 하자. (k+1)번째 프로젝트들의 i번째 속성값과 j번째 속성값들을 모두 고려하여, (j번째 속성값)/(i번째 속성값)의 비율이 낮은 순으로 정렬하여 아래의 그림처럼 효율곡선(Efficient Curve)을 작성한다. i번째 긍정적 속성의 하한값, $LB(pa^i)$ 에 해당하는 효율곡선의 값이 j번째 부정적 속성의 상한값, $UB(na^j)$ 보다 크다면, 현재의 조합은 i번째 긍정적 속성의 하한값과 j번째 부정적 속성의 상한값을 동시에 만족시킬 수 없다.

<그림 2>에서 빗금 친 부분이 긍정적 속성과 부정적 속성 모두를 만족시키는 영역이다. (k+1) 이후 프로젝트들의 선정/비선정의 모든 조합은 효율곡선 위쪽에 놓이게 되므로, 만약 효율곡선이 빗금 친 영역 위에 존재하면 (k+1) 이후 프로젝트들

1) 이를 자세히 설명하면 아래와 같다(J=6, k=3이며, 개선규칙 1이 적용된다고 가정한다). 만약 현 조합에서 개선규칙 1이 적용되면, 3번째 프로젝트의 선정/비선정 여부가 변경되기 전까지 총 8개의 조합은 모두 개선규칙 1이 적용되어 검토할 필요가 없다.

과제 인덱스						개선규칙 1
1	2	3	4	5	6	
1	0	1	0	0	0	적용 <= 현 조합
1	0	1	0	0	1	적용
1	0	1	0	1	0	적용
1	0	1	0	1	1	적용
1	0	1	1	0	0	적용
1	0	1	1	0	1	적용
1	0	1	1	1	0	적용
1	0	1	1	1	1	적용
1	1	0	0	0	0	?

의 선정/비선정을 아무리 변화시켜도 두 개의 제약식을 동시에 만족시키는 조합은 존재하지 않는다.



〈그림 2〉 긍정/부정 속성의 효율곡선

개선규칙 2는 긍정적인 속성과 부정적인 속성의 쌍(pair)마다 적용된다.

4.4 개선규칙 3

세 번째 개선규칙은 모든 속성들이 프로젝트의 속성들에 비하여 좋지 않은 열등 프로젝트에 관한 것이다. 단일 프로젝트 선정 문제에서는 열등 프로젝트는 검토 단계에서 제거하는 것이 마땅하지만, 각 속성들의 상/하한값이 존재하는 포트폴리오 구성 문제에서는 열등 프로젝트가 비열등 포트폴리오에 포함될 수도 있다. 단지 열등 프로젝트가 선정되고 우월한 프로젝트가 선정되지 않는다면 그 포트폴리오는 열등 포트폴리오이므로, 검색단계에서 열등 프로젝트가 선정되고 우월한 프로젝트가 탈락되면 그 조합의 검색을 중단한다.

프로젝트의 수가 J 인 임의의 조합이 있다고 가정하자. 프로젝트 A 는 모든 속성 면에서 프로젝트 B 에 비해 열등하다고 가정하자. 현재의 조합에서 프로젝트 A 가 선정되고 프로젝트 B 가 선정되지 않았다면 현재의 조합은 열등 조합이므로 검토할 필요가 없다.

5. 모의 실험 및 결과

본 연구에서는 개발된 알고리즘의 효율성을 검토하기 위해 모의 데이터를 만들어 하위문제를 탐색하는 데 걸리는 연산시간을 비교하여 보았다. 알고리즘은 C로 작성하였고 Pentium III 1 Giga Herz의 PC를 연산에 사용하였다.

- 모의 데이터의 파라미터는 다음과 같다.

프로젝트의 수	30, 40, 50, 60
각 속성값	일양분포로서 하한값 10, 상한값 100사이의 정수값
상호 배타적 집합	각 포트폴리오마다 1개씩; 프로젝트 3개 중 하나만 선택

본 논문에서 제시한 개선규칙들의 효율성을 검증하기 위해 프로젝트의 수가 30인 10개의 하위문제와 프로젝트의 수가 40인 10개의 하위문제를 설정하여 두 가지 방식으로 연산시간을 측정하여 보았다. 첫 번째 방식은 개선규칙 1만을 사용하였고, 두 번째 방식에서는 개선규칙 1, 2 및 3을 모두 사

〈표 3〉 실험결과 I

번호	J = 30		J = 40	
	A*	B*	A	B
1	218	0	3,600	1
2	118	0	3,600	1
3	35	0	3,600	2
4	118	0	3,600	0
5	122	0	3,600	1
6	78	0	3,600	1
7	108	0	3,600	0
8	122	1	3,600	2
9	55	0	3,600	0
10	39	0	3,600	4
평균	101.3	0.1	3,600	1.2
표준편차	51.0	0.3	N/A	1.2

주) * 표에서 A는 개선규칙 1만을 적용한 경우를, B는 개선규칙 1, 2 및 3을 모두 적용한 경우를 나타낸다.

용하였다. 개선규칙을 전혀 사용하지 않은 방식은 연산시간이 너무 많이 소요되는 관계로 생략하였다. 모의실험의 결과를 아래의 표에 정리하였다. 연산에 적용된 CPU 제한 시간은 3,600초로서 제한시간을 초과한 경우에는 강제로 연산을 종료하였다.

프로젝트의 수가 30개인 경우 개선규칙 1만을 적용하면 평균 연산시간이 101초였으며, 개선규칙을 모두 적용하면 평균 연산시간은 0.1초였다. 프로젝트의 수가 40인 경우 개선규칙 1만을 적용시 10개의 하위문제 중 단 하나도 3,600초 이내에 연산이 종료되지 않았으며, 개선규칙을 모두 적용하면 평균 1.2초 만에 연산이 종료되었다.

개선규칙 1, 2 및 개선규칙 3을 모두 적용할 때, 프로젝트의 수가 증가함에 따라 연산시간이 얼마나 증가하는지를 알아보기 위해 프로젝트의 수를 50 및 60으로 증가시켜 보았다.

〈표 4〉 실험결과 II

번호	J = 50	J = 60
1	267	185
2	3	182
3	69	223
4	8	252
5	13	605
6	2	291
7	31	466
8	37	19
9	20	1790
10	87	58
평균	53.7	407.1
표준편차	76.0	489.8

프로젝트의 수가 50개로 증가한 경우 평균 연산시간은 53.7초였으며, 60개인 경우에는 평균 연산시간이 407.1초였다.

본 연구에서 제시한 알고리즘은 사용자가 포트폴리오의 상/하한값을 조정하고 그 때마다 주어진 상/하한값을 고려하여 비열등 포트폴리오를 찾아나가는 것이므로, 비열등 포트폴리오를 찾아내는

데 소요되는 연산시간은 매우 중요하다. 모의실험의 결과, 본고에서 제시한 개선규칙들은 매우 효율적인 것으로 보이며 프로젝트의 수가 50개 내외까지는 안정적으로 본고에서 제시한 최적기법을 사용할 수 있을 것으로 판단된다.

6. 결 론

본고에서는 R&D 프로젝트 포트폴리오 선정문제를 다루었다. 이 문제가 지니고 있는 의미는 매우 크지만, 이에 대한 실용적 기법은 의외로 극히 적다. 저자들이 아는 한, 체계적으로 비열등 포트폴리오를 구성하는 방법론 중 프로젝트의 수가 30개를 초과하여도 적용할 수 있는 방법론은 학계에 보고되지 않았다. 원인은 이 문제의 특성, 즉 다속성을 지닌 선정문제라는 것에 기인한다. 이런 부류의 문제는 속성 간의 중요도를 어떻게 부여할 것인가라는 어려움과 또한 선정 문제 자체가 지닌 연산상의 어려움을 동시에 지니고 있다. 본고에서는 이러한 어려움을 극복하기 위해 의사결정자와 시스템이 인터페이스(interface)하는 알고리즘을 개발하였다. 의사결정자는 각 속성에 대한 상/하한값을 제시하고 시스템은 그러한 제약을 만족시키는 비열등 프로젝트 포트폴리오를 찾아낸다. 제시된 결과에 의사결정자가 만족하지 않으면 의사결정자는 상/하한값을 조정하고 시스템은 새로운 비열등 포트폴리오를 구성한다. 또한 본 연구에서는 제약조건을 충족시키는 비열등 포트폴리오를 빠른 시간 내에 찾아내기 위해 연산 시간을 단축시키는 3가지 기법을 도입하였다. 모의 데이터를 생성하여 실험한 결과 3가지 기법의 효율성이 매우 높은 것으로 입증되었다.

참 고 문 헌

- [1] Cooper, R.G. and S.J. Edgett, Portfolio New Product Management, McMaster University, Toronto, Canada, 1997.

- [2] Cooper, R.G., "Portfolio Management in New Project : Lessons from Leaders-I," *Research Technology Management*, Vol. 40, No.5(1997), pp.16-28.
- [3] Cooper, R., S. Edgett. and E. Kleinschmidt, "New Problems, New Solutions : Making Portfolio Management More Effective," *Research Technology Management*, Vol.43, No.2(2000), pp.18-33.
- [4] Ghasemzadeh, F. and N.P. Archer, "Project Portfolio Selection through Decision Support," *Decision Support System*, Vol.29(2000), pp.73-88.
- [5] Hess, S.W., "Swinging on the Branch of a Tree : Project Selection Applications," *Interface*, Vol.23(1993), pp.5-12.
- [6] B. Jackson, "Decision Methods for Selecting a Portfolio of R&D Projects," *Research Technology Management*, Vol.26, No.5(1983), pp.21-26.
- [7] Loch, C.H., M.T. Pich, C. Terwiesch, and M. Urbshat, "Selecting R&D Projects at BMW : A Case Study of Adopting Mathematical Programming Models," *IEEE Transactions on Engineering Management*, Vol. 48, No.1(2001), pp.70-80.
- [8] Lieb, E.B., "How Many R&D Projects to Develop?," *IEEE Transactions on Engineering Management*, Vol.45, No.1(1998), pp. 73-77.
- [9] Masood, A.B., D. Davis, and D. Davis, "A Comprehensive 0-1 Goal Programming Model for Project Selection," *International Journal of Project Management*, Vol.19 (2001), pp.243-252.
- [10] Mathieu, R.G. and J.E. Gibson, "A Methodology for Large Scale R&D Planning Based on Cluster Analysis," *IEEE Transactions on Engineering Management*, Vol. 30, No.3(1993), pp.283-291.
- [11] Ringuest, J.L., S.B. Graves and R.H. Case, "Conditional Stochastic Dominance in R&D Portfolio Selection," *IEEE Transactions on Engineering Management*, Vol.47(2000), pp. 478-484.
- [12] Santhanam, R. and J. Kyparisis, "A Multiple Criteria Decision Model for Information System Project Selection," *Computers Operations Research*, Vol.22, No.8(1995), pp.807-818.
- [13] Schmidt, R.L. and J.R. Freeland, "Recent Progress in Modeling R&D Project-Selection Processes," *IEEE Transactions Engineering Management*, Vol.39(1992), pp. 189-199.
- [14] Schniederjans, M. and R. Santhanam, "A Multi Objective Constrained Resource Information System Project Selection Method," *European Journal of Operational Research*, Vol.70(1993), pp.244-253.
- [15] Souder, W.E., "A System for Using R&D Project Evaluation Methods," *Research Technology Management*, Vol.21, No.5(1978), pp.29-37.
- [16] Stewart, T.J. "A Multi-Criteria Decision Support System for R&D Project Selection," *Journal Of the Operational Research Society*, Vol.42(1991), pp.17-26.
- [17] Stummer, C. and K. Heidenberger, "Interactive R&D Portfolio Selection Considering Multiple Objectives, Project Interdependencies, and Time : A Three-Phase Approach," *Management of Engineering and Technology*, (2001), pp.423-428.
- [18] Taylor, B., L. Moore and E. Clayton, "R&D Projection Selection and Manpower Allocation with Integer Nonlinear Goal Programming," *Management Science*, Vol.28, No.10 (1992), pp.1148-1158.