Journal of the
Military Operations Research
Society of Korea, Vol. 5, No. 2
December, 1979

A Project Selection Model Allowing Resource Flexibility

Chung, Joon Wha *

要 約

本 論文은 体系的인 演算法(Algorithm)을 研究, 開発하여 予算算出에 重要한 役割을 하는 mixed integer programming 問題를 計算하기 為하여

첫째로, 많은 投資手段 또는 計劃中 가장 合理的인 (利益率이 높은) 手段을 定하는 Weingartner의 予算算出모델(Capital Budgeting Model)을 簡単히 検討해 보고, 둘째로 위의 検討에서 얻어진 槪念에 따른 이 모델의 問題点들을 摸索해 보며 마지막으로 企劃分野의 어떤 期間동안 資源確保(拡大)를 可能케 하는 予算算出 問題를 解決할 수 있는 方案이나 体系的 演算法을 開発해 보고자 한다. 이러한 体系的 演算法은 (含蓄的인 또는 部分的인) 列挙方式(Enumeration Technique)을 利用함으로써 위에 提示된 問題를 解決하는데 容易할 것이다.

I. OBJECTIVES

This paper is intended to develop an algorithm for solving a mixed integer programming (binary) in dealing with capital budgeting problems. First, the Weingartner's capital budgeting model which deals with the problem of allocating fixed budgets among competing investment proposals is reviewed briefly. Next, we discuss some problems inherent in the model along with some insight gained from the above review. Last, we try to develop some heuristic rules or an algorithm to solve the capital budgeting problem allowing some extension of the resources in each time period of the planning horizon.

* 陸士 經濟學科

By utilizing the implicit (or partial) enumeration technique, the solution algorithm is to utilize some special features existing in such problems.

II. MOTIVATION

In dealing with the problem of allocating fixed budget dollars among competing investment proposals, Weingartner[1] suggested an LP approach to this problem. Since decisions about individual projects must be made on an all-or-nothing basis due to the indivisibility of investment projects, integer programming (ILP) method can be used to deal rigorously with the indivisibility of investment projects. However, when there are so many projects to consider, the computer can not handle a large number of variables. Therefore, the Weingartner's LP method has some advantages over the usual ILP due to the simplicity in computerbility and interpretation.[2] The Weingartner's model can be stated as follows:

$$\text{Maximize} \sum_{j=1}^{n} b_j X_j$$

$$\text{subject to} \sum_{j=1}^{n} c_{tj} X_j \leq C_t \quad , \quad t = 1, \ldots\ldots\ldots, m$$

$$0 \leq X_j \leq 1$$

where c_{tj} = costs of projects

C_t = budget ceilings in year t

b_j = present value of all revenues and costs associated with individual projects

X_j = decision variables (0, 1, or the fraction of project j undertaken)

The Weingartner's LP model with the constraint $0 \leq X_j \leq 1$ requires n 1's in the values of X_j (decision variables) or slacks q_j in the final solution (n=the number of possible projects). With t more constraints (t=the number of years constrained by budget ceilings), the total number of constraints is (n+t). In an LP problem, the maximum number of non-zero solutions is equal to the number of independent vectors (rank); hence, there is no way to satisfy the n constraints without either X_j or q_j being zero or one. Therefore, the remaining to solutions should be either one's in some cases or zeroes. Unless we have very many years to consider, the number of t years to be constrained will not be very many; hence, the number of fractional solutions are not likely to be numerous. In addition, the truncation effects due to considering a relatively short time horizon can offer some protection against the uncertainties in long-range forecasts.[3] Each fractional project per period can be interpreted as directions to try a smaller version of the same project or can be eliminated completely by using integer programming. Since in integer programmings, only binary variables will be involved in this type, an optimal solution may be obtained with some resources remaining unused as in strict LP models. Contrary to Weingartner's problems of fractional solutions, however, we may allow the resource constraints to vary within some reasonable ranges. Thus, in strictly binary ILP, those projects not previously accepted due to the violation of integer constraints can be included in the final solution by allowing some additional resources. Of course, in this case we have to consider the trade-offs between the amount of benefits resulted from adding some more projects and the penalty costs associated with borrowing some additional funds. The objective is to add to the net present value of income streams, thereby maximizing the total present values of the investment projects. We want to find out the optimal value of additional resource requirements which will more than offset the needed penalty costs due to the increased resources.

III. PROBLEM FORMULATION

A firm has n projects that it would like to undertake but, because of budget limitations, not all can be selected. In particular, project j has a present value of C_j, and requires an investment of a_{ij} dollars in the time period i, i=1, ..., m. The capital available in time period i is b_i. The problem of maximizing the total present value subject to the budget constraints can be written as:

$$\text{Maximize} \sum_{j=1}^{n} C_j X_j$$

$$\text{Subject to} \sum_{j=1}^{n} a_{ij} X_j \leq b_i, \quad i=1,\ldots\ldots,m$$

$$X_j = 0,\ 1, \quad j=1,\ldots\ldots,n$$

where X_j = 1 if project is selected or X_j = 0 otherwise. The above is a usual capital budgeting problem formulated in terms of integer programming Now we wish to further consider the possibility of relaxing some resource constraints as variables. P_i is the present value of the penalty costs associated with borrowing some additional funds or the present value of the opportunity costs of additional capital in time period i. The reformulated model with some resource flexibility allowed can be written as follows:

$$\text{Maximize} \sum_{j=1}^{n} C_j X_j - \sum_{i=1}^{m} P_i Y_i$$

$$\text{subject to} \sum_{j=1}^{n} a_{ij} X_j \leq b_i + Y_i, \quad i=1,\ldots\ldots,m$$

Y_i denotes possible additional resources available for each period

$$X_j = 0,\ 1 \quad \text{for all } i$$

$$Y_i \geq 0$$

In dealing with a maximization problem, it is possible to rearrange such problem so that $C_j \leq 0$ by making the substitution $X_j = 1 - \overline{X}_j$ for $C_j > 0$, where $\overline{X}_j$ is the complement of X_j in the constraint $X_j \leq 1$. By doing this, we assure oruselves that a basic dual feasible solution, which is used as a starting point, is available.[4] The entries C_j and a_{ij} are not required to be integer. The model is a mixed integer program (MILP).

Special Features of the Problem

Whenever the left handside of the constraint is less than or equal to b_i, the value of Y_i is zero. On the other hand, whenever the left handside of the constraint is greater than b_i, a value is assigned to Y_i, thereby having the constraint be an equality constraint. By taking advantage of the model's special feature mentioned above, we can solve the problem using implicit enumeration technique. Before tackling the algorithm for the above problem, some preliminary discussion of the notation to be used is in order. This notation is based on Garfinkel, Robert S., and Nemhauser, George L., Integer Programming, John Wiley & Sons, New York, 1972, pp. 122-127.

Separation

Given $$S_0 = \{ X \mid AX \leq b,\ x \text{ binary} \}$$

the separation at v_k is determined by choosing a particular variable x_j not chosen previously along the path P_k from v_0 to v_k, and letting

$$S_k^* = \{ S_k \cap \{ x \mid x_j = 0 \},\ S_k \cap \{ x \mid x_j = 1 \} \}$$

The path P_k corresponds to an assignment of binary values to a subset of the variables. Such an assignment is called a partial solution. Denote the index set of the assigned variables by W_k and let

$$S_k^+ = \{ j \mid j \in W_k \text{ and } x_j = 1 \}$$

$$S_k^- = \{ j \mid j \in W_k \text{ and } x_j = 0 \}$$

$$F_k = \{ j \mid j \notin W_k \}$$

A completion of W_k is an assignment of binary values to the free variables specified by the index set F_k.

Bounding

The problem considered at v_k (vertex) is

$$\max z_k = \sum_{j \in F_k} c_j x_j + \sum_{j \in S_k^+} c_j$$

$$\sum_{j \in F_k} a_{ij} x_j \leq b_i - \sum_{j \in S_k^+} a_{ij} = s_i, \quad i = 1, \ldots\ldots, m$$

$$x_j = 0, 1, \; j \in F_k$$

Since $c_j \leq 0$, $x^0(k)$ is obtained by setting $x_j = 0$, $j \in F_k$. Thus $\bar{z}_k = z_k^0 = \sum_{j \in S_k^+} c_j$. If, in addition $s = (s_1, \ldots\ldots, s_m) \geq 0$, then $x^0(k)$ is feasible and $\underline{z}_k = z_k^0$.

Fathoming

The fathoming cases are (a) $\bar{z}_k = \underline{z}_k$ (b) $\bar{z}_k \leq \underline{z}_0$.

Some Additional Notations

$N_k = \{ i \mid s_i < 0 \}$

$P_k = \{ P_1^k , P_2^k , \ldots\ldots\ldots, P_j^k , \ldots P_t^k \}$ where P_j^k is for X_j for $j \in F_k$ and in the increasing order of C_j / r_j where $r_j = \sum_{i=1}^{m} a_{ij}$

According to Zionts,[5] the ordering of variables in branch and bound methods can be particularly important, because many of the methods branch on the "next" free variable that is required to be integer. Therefore, zero-one variables should appear, from left to right, in order of increasing costs for a minimizing problems or in order of decreasing profits for a maximizing problem.

Since in our previous formulation of the problem, we can assume that c_j and a_{ij} are positive or better nonnegative and we are dealing with maximization problems, we can state the problem in the following way, by replacing $x_j = 1 - \overline{X}_j$.

$$\text{Max} \; -\sum_{j=1}^{n} c_j X_j + \sum_{j=1}^{n} c_j - \sum_{i=1}^{m} P_i Y_i$$

$$\text{subject to} \; -\sum_{j=1}^{n} a_{ij} x_j \leq b_i - \sum_{j=1}^{n} a_{ij} + Y_i$$

IV. ALGORITHM

Step 1) At v_0, initialize $\underline{z}_0 = -\infty$ and $\overline{z}_0 = +\infty$. Partition on X_p where p is the first element of P set. Go to Step 2.

Step 2) At V_k, if s 0, let $\underline{Z}_k = -\sum_{j \in S_k^+} C_j$. Go to Step 3. Otherwise, set $Y_i = -s_i$ for i N_k and set the other $Y_i = 0$ and let $\underline{z}_k = \sum_{j \in S_k^+} C_j$ $- \sum_{i \in N_k} P_i Y_i$. Go to Step 3.

Step 3) If $\underline{z}_k \geq \underline{z}_0$, save the index of that vertex and the corresponding values for x and set $\underline{z}_0 = \max \{ \underline{z}_0 , \underline{z}_k \}$, and go to step 6. Otherwise, go to Step 4.

Step 4) Fathom that vertex. Go to Step 5.

Step 5):(Backtracking): If no live vertex exists, go to Step 7. Otherwise, branch to the newest live vertex and go to Step 6.

Step 6):(Partitioning and Branching): Partition on X_p where p is the first element of P_k set and renew the P_k set. Branch to the $X_p = 1$ vertex. Go to Step 2.

Step 7):(Termination): If $\underline{z}_0 = -\infty$, there is no feasible solution. If $\underline{z}_0 > -\infty$, the solution is optimal.

Some Additional Considerations

In order to make the model more realistic, we can incorporate into the model the following additional constraints:

(1) $0 \leq Y_i \leq M_i$

In order to guarantee bounded solutions, we can set the upper limit on the maximum available amount for Y_i which is also reasonable in real world situations.

(2) $\sum_{r \in L_k} X_r \leq 1$

To handle mutually exclusive projects, we simply add the above constraints.

(3) $X_a \leq X_b$

If project a is desirable only if project b is adopted, but not otherwise, then the above constraint will handle the problem.

In addition, we can incorporate into the model some additional constraints to be checked at one of the stages: that is, shifting funds between periods, shifting projects between periods, inclusion of other bottlenecks into the model, etc. (See Quirin).

V. AN EXAMPLE

Suppose that we want to maximize the following subject to the constraints.

$$\text{Maximize } 500X_1 + 450X_2 + 400X_3 + 200X_4 - 5Y_1 - 2Y_2$$

$$\text{Subject to } 400X_1 + 300X_2 + 350X_3 + 300X_4 \leq 1{,}000 + Y_1$$

$$300X_1 + 200X_2 + 400X_3 + 450X_4 \leq 900 + Y_2$$

$$X = 0 \text{ or } 1,\ Y \geq 0$$

Since the above is a maximization problem, we replace X_j by $(1 - \bar{X}_j)$ and rewrite the above as follows:

$$\text{Maximize } -500\bar{X}_1 - 450\bar{X}_2 - 400\bar{X}_3 - 200\bar{X}_4 - 5Y_1 - 2Y_2 + (1{,}550)$$

$$\text{Subject to } -400\bar{X}_1 - 300\bar{X}_2 - 350\bar{X}_3 - 300\bar{X}_4 \leq -350 + Y_1$$

$$-300\bar{X}_1 - 200\bar{X}_2 - 400\bar{X}_3 - 450\bar{X}_4 \leq -450 + Y_2$$

Solution Algorithm

(1 - Step 1) $\underline{Z}_0 = -\infty$, $\bar{Z}_0 = +\infty$ Choose X_4 from the P set where the P set is in the increasing order of 200/750, 400/750, 500/700, and 450/500. (The order of X_j is then X_4, X_3, X_1, and X_2)

(1 - Step 2) At V_1, check whether s = 0 or not.

$$\text{Max } -500\bar{X}_1 - 450\bar{X}_2 - 400\bar{X}_3 - 200 - 5Y_1 - 2Y_2$$

$$\text{Subject to } -400\bar{X}_1 - 300\bar{X}_2 - 350\bar{X}_3 \leq -50 + Y_1$$

$$-300\bar{X}_1 - 200\bar{X}_2 - 400\bar{X}_3 \leq 0 + Y_2$$

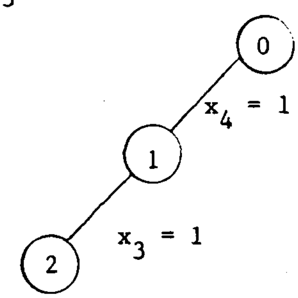
Since s_i is not all ≥ 0 ($s_1 = -50$ and $s_2 = 0$), set $Y_1 = -s_1$, $Y_2 = 0$.

Then at V_1, $\underline{Z}_1 = -200 - 5(50) = -450$.

(1 - Step 3) Since $-450 = \underline{Z}_1 > \underline{Z}_0 = -\infty$, set $\underline{Z}_0 = -450$

Go to Step 6.

(1 - Step 6) Choose X_3 (the next one in the P set) and branch to $X_3 = 1$. Go to Step 2.



$$\text{Max } -500\bar{X}_1 - 450\bar{X}_2 - 5Y_1 - 2Y_2 - 600$$

Subject to

$$-400\bar{X}_1 - 300\bar{X}_2 \leq 300 + Y_1$$

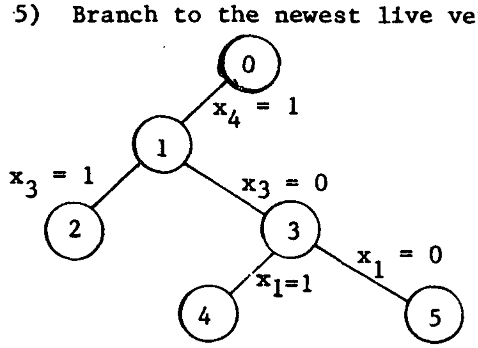
$$-300\bar{X}_1 - 200\bar{X}_2 \leq 400 + Y_2$$

(2 - Step 2) Since $s = \begin{bmatrix} 300 \\ 400 \end{bmatrix} > 0$, $\underline{Z}_2 = -\sum_{j \in S_k} C_j = -600$. Go to Step 3. (Here we have not considered the (1,550) which is the upper bound in this case and a constant)

(2 - Step 3) Since $-600 = \underline{Z}_2 < \underline{Z}_0 = -450$, go to Step 4.

(2 - Step 4) Fathom and go to Step 5.

(2 - Step 5) Branch to the newest live vertex and go to Step 6.



(3 - Step 6) Choose X_1 and go to Step 2.

(4 - Step 2) At V_4, see if the s set ≥ 0 or not.

Max $-500 - 450\bar{X}_2 - 200 - 5Y_1 - 2Y_2$, which is reduced to

$-450\bar{X}_2 - 700 - 5Y_1 - 2Y_2$, subject to

$-300\bar{X}_2 \leq 350 + Y_1$

$-200\bar{X}_2 \leq 350 + Y_2$

Since the set $s = \begin{pmatrix} 350 \\ 350 \end{pmatrix} > 0$, $\underline{Z}_4 = -700$. Go to Step 3.

(4 - Step 3) Since $\underline{Z}_4 < \underline{Z}_0$, go to Step 4.

(4 - Step 4) Fathom and go to Step 5.

(5 - Step 5) Branch to the newest live vertex and go to Step 6.

(5 - Step 6) Choose X_2 and branch to $X_2 = 1$ vertex. Go to Step 2.

Max $-650 - 5Y_1 - 2Y_2$

subject to $-600 \leq -350 + Y_1$

$-650 \leq -450 + Y_2$

(6 - Step 2) Since $s > 0$, $\underline{Z}_6 = -650$ and go to Step 3.

(6 - Step 3) Since $\underline{Z}_6 < \underline{Z}_0$, go to Step 4 and fathom that vertex.

Go to Step 5. Branch and go to Step 6.

At V_7, we have $\underline{Z}_7 = -450$. Then this vertex is fathomed.

(9 - Step 2)

Max $-500\bar{X}_1 - 450\bar{X}_2 - 400 - 5Y_1 - 2Y_2$

subject to $-400\bar{X}_1 - 300\bar{X}_2 \leq 0 + Y_1$

$-300\bar{X}_1 - 200\bar{X}_2 \leq -50 + Y_2$

In the above, since $s = \begin{pmatrix} 0 \\ -50 \end{pmatrix} \leq 0$, $Y_1 = 0$ and $Y_2 = 50$.

At V_9, $\underline{Z}_9 = -400 - 2(50) = -500$ which is smaller than $\underline{Z}_0 = -450$. After checking the possible $X_j = 0$ branch, we have found the value at V_1 is the largest, that is, the optimum value is -450, where $X_4 = 1$ and $Y_1 = 50$. Therefore, choose the projects X_1, X_2, and X_3 but not X_4.

In addition, allow the additional amount of $Y_1 = 50$.

The above technique so far described checks all the $X_j = 0$ branches but not all $X_j = 1$ branches. This fact gives rise to the advantages of partial enumerations.

REFERENCES

1. Weingartner, H.M., Mathematical Programming and the Analysis of Capital Budgeting Problem, Englewood Cliffs, N.J.: Prentice-Hall, 1963, pp. 16-38.

2. Ibid., p. 16.

3. Quirin, G. David, The Capital Expenditure Decision, Chapter 9, pp. 187-188.

4. Zionts, Stanley, Linear and Integer Programming, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1974, p. 438.

5. Ibid., p. 480