Optimization of Capital Budgeting for Interrelated Capacity Expansion Projects

ALI E. HAGHANI AND CHIEN-HUNG WEI

A network representation procedure for the optimization of capital budgeting problems is proposed. Interrelated capacity expansion projects and discrete time decisions are considered. An alternative-based approach is proposed that yields a very neat formulation by identifying the relationships among alternatives before solving the problem. The most general pairwise interactions are considered, and a 0-1 integer program is formulated with nonlinear objective function and linear constraints. Procedures are demonstrated that convert the nonlinear program into a linear form with an embedded network structure. The out-of-kilter algorithm is applied to solve the resulting network flow problem. Because of the inherent nature of the network structure, this approach enables efficient calculation of optimal solutions for relatively large problems. Computational efficiency is shown that requires less than 3 min of central processing unit time for a system with 372 alternatives.

Capital budgeting problems are of interest in many disciplines. Each field shows unique perspectives and priorities and utilizes different tools and techniques. Because of the application of specific characteristics, the literature on budgeting has tended to divert from, rather than converge toward, a unified perspective. Basically, capital budgeting deals with the evaluation, selection, sequencing, and scheduling of investment projects. Consideration of interrelated projects is particularly challenging because the input or output factors, or both, of one project are significantly affected in magnitude or timing, or both, by the selection or rejection decisions on one or more of the others under consideration. In real-world cases, projects tend to affect each other in terms of costs or benefits. Therefore, consideration of interrelated projects will provide decision makers with more precise information.

The evaluation of projects is mainly an economic issue that heavily depends on the information available about techniques, markets, demands, predicted costs, and benefits, whereas project selection, sequencing, and scheduling basically make up an optimization process. The evaluation stage is often separated from other stages because of different characteristics. Therefore, the objective as well as the scope of this paper, given the full information on project evaluation, is to develop an efficient procedure for making decisions among interrelated projects to meet prespecified goals and objectives over a planning horizon. By applying the proposed procedure, the three major stages in capital budgeting, that is, project selection, sequencing, and scheduling, could be fulfilled in a single stage.

This paper is organized as follows. The problem is described in more detail and a brief literature review is presented in the next section. Then the problem is formulated as a mathematical optimization problem and the solution approach is provided. Numerical results are also presented. Finally, conclusions and future research are discussed.

Department of Civil Engineering, University of Maryland, College Park, Md. 20742.

PROBLEM STATEMENT

There has been much research concerning interrelated project scheduling. Various fields have a similar problem, and the study methods are diversified [e.g., Erlenkotter (*1*), Gear and Cowie (*2*), Luss (*3*), Czajkowski and Jones (*4*), and Gomes (*5*)]. Most approaches to capital budgeting use integer programming, dynamic programming, or multicriteria formulation to model the interdependencies among projects. To overcome the combinatorial difficulty when more interrelated projects are involved, various heuristic methods are developed. However, studies on project relationships have not yet exhibited a significant breakthrough. For example, the study by Srinivasan and Kim (*6*) was not quite applicable to practical problems.

The solution space of the capital budgeting problems grows exponentially when more projects are considered. However, in most real-world cases, not all of the projects are pairwise interrelated. Hence, decomposing the project set into smaller subsets shows some advantage. Decomposition may be done according to specific applications or by following the more rigorous rules shown by Steuer and Harris (*7*) or Morse (*8*). Prescreening projects to reduce the problem size should be considered whenever possible.

Most of the existing techniques use relatively small illustrative examples. It is thus difficult to assess their capabilities in dealing with large problems. Heuristic approaches, for example, that of Janson et al. (*9*), are developed to handle such large problems. These procedures attempt to reduce the number of projects under consideration systematically or to find "good" solutions to the problems, or both. However, in most cases, it is not possible to verify the quality of the heuristic solutions relative to the optimal solutions. This is because the heuristic procedures usually lack a built-in mechanism to evaluate the quality of each solution obtained. The approach proposed in this paper attempts to overcome such limitations. The proposed model and solution procedure enable the analyst to handle relatively large problems easily and to find the optimal solutions with relatively minor computational efforts.

The capacity expansion problem is similar in many respects to the general project sequencing problem that determines the sizes and facilities to be added. They both are fairly difficult to solve (*10*). A special class of capacity expansion deals with a set of expansion projects according to the demand pattern over time. Then the problem is to select the sequence and appropriate timing of projects to satisfy the time-dependent demand while minimizing the total discounted costs. The resulting expansion path may be defined as the order and timing of implementing a subset of preferred projects. In the work of Martinelli (*11*), a graphical depiction was proposed to sequence inland waterway capacity expansion projects. Each project was viewed as a system generating a common time-dependent output, and a two-dimensional representation was utilized in which costs are plotted on the vertical axis and time is plotted on the horizontal axis. A search algorithm is required in selecting the preferred sequence of projects in the corresponding two-dimensional space. The expansion path could then be identified by the implementation sequence of projects and associated times. This method, named continuous time expansion for convenience, could be a powerful support for decision making because of its visual structure. The expansion paths, however, may not be easily and correctly recognized instantly through the graph, especially for complex project combinations. Hence, direct application of this method may be limited to a small number of projects. Consequently, Martinelli developed a heuristic sequencing algorithm to obtain an efficient solution.

To achieve a sequential expansion path using Martinelli's search algorithm, the cost functions were hypothesized to possess desired properties. Specifically, positive first and second derivatives were assumed with respect to the system outputs. Although it holds in waterway systems, such a requirement might be another limitation of the continuous time approach to more general applications. This type of application-specific restriction is one of the urgent issues to be resolved in capital budgeting.

A good solution for capital budgeting may be easily found by slightly twisting the original problem. Wei (*12*) developed an approach based on discrete time decisions. In that work, the choice of individual projects was the primary decision variable, and a set of supplemental variables was needed to identify the combinations of projects. Because the interrelations among projects are reflected in the cost functions of various project combinations, the transitions between projects and combinations of projects need to be incorporated explicitly as constraints. Consequently, the resulting objective function is quite complicated. That formulation was solved exactly using LINDO.

A comparison between the discrete and continuous time approaches reveals that these two approaches may eventually generate similarly good results for investment purposes. Figure 1 shows a simple example, in which two projects are considered. A total number of three alternatives and corresponding cost functions are identified in the figure. The optimal continuous time decision will start Project 1 at the beginning, switch to 1 + 2 at Year 2.3, and keep on 1 + 2 until the end with a total of 73.48 units. The optimal discrete time decision is to implement Project 1 for the first year and add Project 2 to the system at Year 2 with a total cost of 74.3 units. The expansion paths are highlighted in Figure 1 for the two different approaches. This example indicates that although it might not yield the optimal result, the discrete time decision approach can provide a total cost that is very close to the optimum and good enough for practical applications.

Another property of the discrete time approach is that the selection and sequencing result will be the same as that obtained in the continuous time approach as long as each discrete period is not too long, for example, 1 year. As shown in the following sections, a network solution procedure can

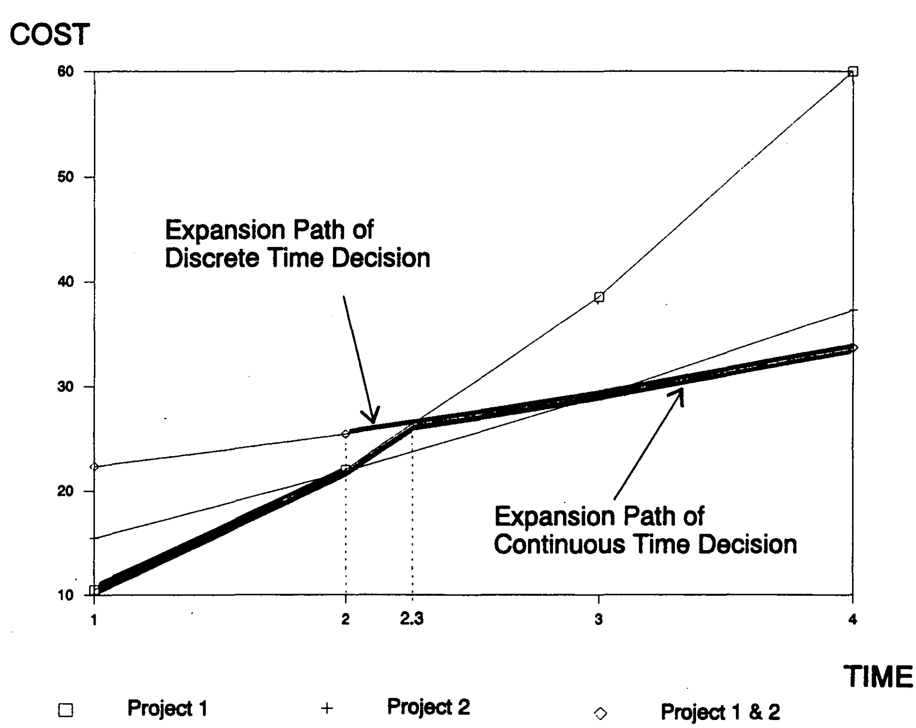


FIGURE 1 Motivating example: comparison of continuous and discrete time project selections.

be applied to the discrete time approach to generate exact solutions.

Because assessing the exact interdependence terms has not been satisfactorily reported in the literature, this paper considers the aggregate effects of interdependent project selection. Therefore, identifying and evaluating interrelations among projects is exogenous to this study, as mentioned earlier. The facility economic service lives are assumed longer than the time horizon of interest. The total discounted costs associated with all combinations of projects are given explicitly as functions of time or demand. These costs include user costs, capital costs, and maintenance costs of projects. The cost functions are usually nonlinear. However, no assumptions are made regarding the functional forms of these costs. It is only assumed that the cost functions are continuous and integrable over the entire planning horizon. Budget constraints are temporarily disregarded. However, they can be incorporated in the model with minor modifications. In fact, considering budget constraints would make the proposed solution method more advantageous, which will be addressed later.

PROBLEM FORMULATION

The problem of choosing projects among candidates for a limited number of periods readily reveals the nature of integer programming. The objective is to minimize total costs while satisfying a certain level of demand. In fact, the example in Figure 1 clearly possesses such a characteristic. For each individual project or combination of projects, it is assumed that implementation may be made either at the beginning of the planning horizon or at the beginning of a later period, depending on the resulting costs. Before formally presenting the formulation, several terms need to be defined. A project combination is referred to as an "alternative." Alternative j is an "increment" of alternative i if every project in alternative i is also in alternative j. Alternatives are mutually exclusive if incremental relationships do not hold. For instance, if three projects are considered, alternative 1 + 2 + 3 is an increment of alternatives 1 + 2, 1 + 3, 2 + 3, 1, 2, and 3, and alternatives 1 + 2 and 2 + 3 are mutually exclusive.

To consider full interdependence among projects, pairwise interactions are assumed among all individual projects. Consequently, a small number of projects will reflect a relatively large set of alternatives; for example, five projects represent the consideration of 31 alternatives. This is the most general case possible. However, in reality the pairwise interrelation usually does not exist between every pair of projects; therefore, the number of alternatives would be much fewer.

The primary constraint considered in this paper is the continuity of projects. For research and development industries, projects may be terminated and entirely removed when the outcome is unsatisfactory [Shafer and Mantel (*13*) or Bard et al. (*14*)]. However, in many circumstances no project can be removed once it is implemented or if the cost associated with termination is extremely high. Specifically, decisions at any time should not cause the abandonment of projects that have been implemented in earlier periods through selection of other alternatives.

Because of the complexity of identifying projects and alternatives, the proposed formulation is entirely alternative based. That is, the pairwise interactions among projects are specified explicitly and are not included in the integer program. Consequently, the model presented below has a very neat form and, as shown in the next section, leads to the efficiency of the exact solution procedure.

Assuming that all projects have positive effects on the system under consideration, the underlying idea of the proposed integer programming is to decide which alternative to implement at each period subject to relevant constraints. This means that for the planning horizon only one alternative is selected in each period. As mentioned earlier, interdependencies among projects and corresponding cost functions are specified exogenously. The alternatives may be arranged, for calculation convenience, by the costs at base time such that the lower-cost alternatives will be assigned smaller indexes. The following definitions are used:

I = set of all alternatives;
H = set of periods in the planning horizon;
W_{it} = 1 if alternative i is implemented in period t, 0 otherwise; and
$f_i(t)$ = cost of alternative i as a function of time.

To minimize the total cost associated with all possible alternatives for the entire horizon, the corresponding values of costs for each alternative and time period need to be computed. These values are equivalent to the shaded area under each curve segment shown in Figure 2. Because the cost functions are usually nonlinear, simple algebra is not applicable. Rather, integrating the cost functions for each period is desired. The binary decision variable W_{it} is required to ensure the best choice among alternatives—one that minimizes the associated costs and fulfills relevant restrictions. Using the notations defined above, the objective function is

$$\text{Minimize} \sum_{i \in I} \sum_{t \in H} W_{it} \int_t^{t+1} f_i(u)du \tag{1}$$

Here the flexibility of various types of cost functions with respect to time or demand level is allowed.

Because projects always generate benefits to the system of concern, it is needed to ensure one alternative selected for each period. That is, at least one project should be in service at any time. We then have the following constraint:

$$\sum_{i \in I} W_{it} = 1 \qquad \forall\, t \in H \tag{2}$$

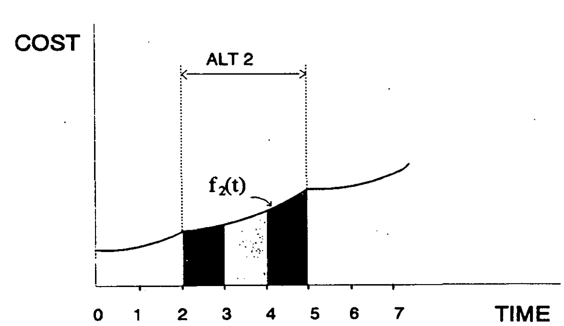


FIGURE 2 Example of optimal expansion path.

Note that Equation 2 also implicitly prevents the conflict of alternatives because one and only one alternative is allowed for each period.

Ensuring the continuity of projects is very important, especially when system capacities are expanded. This condition should hold throughout the entire horizon such that, if alternative i is implemented in period t_0, alternative j must be an increment of alternative i to be implemented in period $t_1 > t_0$. The incremental relationship is easily specified before the formulation. Let J_i be the set of increment alternatives of alternative i. The following constraint satisfies this requirement:

$$W_{it} \leq \sum_{j \in Ji} W_{j,t+1} \qquad \forall\, i \in I, t \in H \tag{3}$$

Constraint 3 states that if a project is implemented in a certain period, it should be included in the alternatives chosen in later periods. This ensures the continuity of all projects in any period and is actually a very large set of inequalities.

The alternative-based integer programming for project scheduling is summarized below:

$$\text{Minimize} \sum_{i \in I} \sum_{t \in H} W_{it} \int_t^{t+1} f_i(u)du$$

Subject to

$$\sum_{i \in I} W_{it} = 1 \qquad \forall\, t \in H$$

$$W_{it} \leq \sum_{j \in Ji} W_{j,t+1} \qquad \forall\, i \in I, t \in H$$

$$W_{it} = 0 \text{ or } 1 \tag{4}$$

The above formulation is a 0-1 integer programming problem with a nonlinear objective function and linear constraints. Note that no specific assumption is made regarding the functional forms of the costs [i.e., $f_i(u)$]. In fact, as one will see shortly, if these functions are continuous and Rieman integrable, no further assumption is necessary.

As stated before, in previous work the relationship between projects and alternatives was incorporated and resulted in a complicated linear formulation requiring much computation time. Consequently, heuristic and approximation methods were employed to save computational effort. In contrast, the above formulation can be modified into a linear program with a network structure that can be solved exactly by several well-known procedures. This advantage is discussed below.

SOLUTION PROCEDURES

In this section, a procedure is outlined for converting the nonlinear program into a linear program. Then it is shown how the linear program can be reformulated as an out-of-kilter network flow problem and the characteristics of the out-of-kilter algorithm (OKA) are briefly described. Finally some moderate-size numerical examples are presented to test the computational efficiency. The LINDO program is also used to solve these examples.

Linear Programming

Instead of solving the nonlinear program (*4*) directly, a much more efficient solution procedure is achievable through minor transformation of the objective function. This step is desirable to take advantage of the inherent network structure.

Since the known cost functions are continuous and integrable, one could compute the integration portion of the objective function before any other manipulations. Let C_{it} be the cost associated with the decision variable W_{it}. Then the cost of implementing alternative i in period t is

$$C_{it} = \int_t^{t+1} f_i(u)du \tag{5}$$

Note that Equation 5 is applicable to any type of cost function. The resulting values of C_{it} may be treated as scalar coefficients associated with W_{it}. Therefore, the objective function is now transformed into a linear form:

$$\text{Minimize} \sum_{i \in I} \sum_{t \in H} C_{it} W_{it} \tag{6}$$

Given that the magnitudes of C_{it} could always be calculated outside the program, the linear program considered here is

$$\text{Minimize} \sum_{i \in I} \sum_{t \in H} C_{it} W_{it}$$

Subject to

$$\sum_{i \in I} W_{it} = 1 \qquad \forall\, t \in H$$

$$W_{it} \leq \sum_{j \in Ji} W_{j,t+1} \qquad \forall\, i \in I, t \in H$$

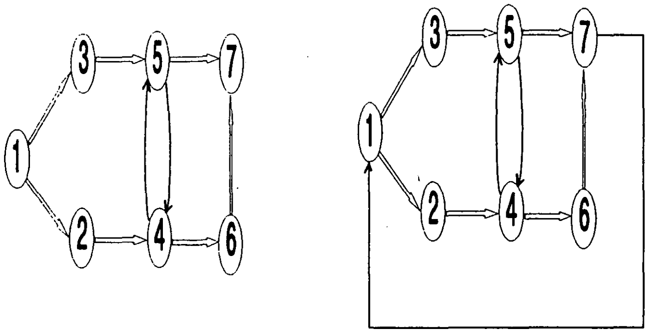
$$W_{it} = 0 \text{ or } 1 \tag{7}$$

Program 7 is found to possess a special structure that could be solved more efficiently by network-flow techniques than the conventional linear programming simplex method. The well-known OKA is demonstrated for this application.

Out-of-Kilter Algorithm

The OKA is developed using the concepts of linear programming duality theory and complementary slackness conditions. The algorithm is especially designed to deal with capacitated network-flow problems. Hence, it is more suitable than the simplex method for linear program problems with network structure.

To apply OKA, a closed network representation is desired that exhibits flow circulation. A circulation is an assignment of flow to arcs such that flow is conserved at each node. The OKA deals with circulations; thus it is often necessary to modify the original networks. A regular network and the corresponding closed network are shown in Figure 3. A return arc is needed from Node 7 to Node 1, as shown in Figure 3*b*. The details of the return arc depend on the problem described by the network.



(a) Original Network (b) Closed Network

FIGURE 3 Network representation.

Associated with each arc in a capacitated network are lower and upper bounds and costs of flow. These may, in fact, be 0 or infinity, as long as constraints are fulfilled. The following notation is defined for OKA:

S = set of arcs,
N = set of nodes,
f_{ij} = flow-through arc (i, j),
L_{ij} = lower bound of flow on arc (i, j),
U_{ij} = upper bound of flow on arc (i, j), and
c_{ij} = cost associated with shipping one unit flow on arc (i, j).

The network-flow problem (or minimum cost circulation problem) is a special linear programming problem that minimizes the total cost subject to four sets of constraints. The general model is

$$\text{Minimize} \sum_{(i,j)\in S} c_{ij} f_{ij}$$

subject to

$$\sum_{j\in N} f_{ji} - \sum_{j\in N} f_{ij} = 0 \qquad \forall\, i, j \in N,\ i \neq j$$

$$f_{ij} \leq U_{ij} \qquad \forall\, (i, j) \in S$$

$$f_{ij} \geq L_{ij} \qquad \forall\, (i, j) \in S$$

$$f_{ij} \geq 0 \qquad \forall\, (i, j) \in S \tag{8}$$

These constraints represent the requirements on flow conservation, upper bound, lower bound, and nonnegativity, respectively.

The OKA is an iterative procedure to find the optimal circulation in a capacitated network characterized by Program 8. The detailed steps have been described by Phillips and Garcia-Diaz (*15*). Although the OKA procedure seems tedious, it is well defined and can be easily computerized. In addition, because the OKA has very wide application, the procedure and the criterion for optimality will not be altered for various problems. The only changes necessary are the associated network configurations.

A residual benefit to the OKA is that the problem can be easily visualized, a property not present in linear programming formulations in more than two dimensions. Hence, properly presenting the problem as a closed-loop capacitated network-flow problem is essentially the key step. To efficiently solve the discrete time expansion problem, Program 7 may be translated into some equivalent problem that certainly could be solved by the OKA. The typical shortest-path problem is suitable for this purpose.

Each node may be treated as a decision variable (i.e., W_{it}) and each arc as a transition from the current decision variable to the next. In the network, the arc costs correspond to implementing alternatives in a given period (i.e., C_{it}). Because the continuity of projects must hold, nodes may be connected with appropriately selected arcs that satisfy this concern. The resulting network is directed because of the incremental relations between alternatives. The restrictions of implementing only one alternative for each period and nonnegative decision variables imply that flow assigned to each arc will be either unity or 0. In other words, the lower bound of flow is 0 and the upper bound of flow is 1 for arcs connecting with decision variables. To accomplish a closed network, a supersource node and a supersink node are desired to represent the beginning and the end of the time horizon. The supersource and supersink nodes are connected to the first and the last period, respectively, and the return arc from supersink to supersource is created accordingly. The return arc has lower and upper bounds of flow both equal to 1, and the cost is 0 to ensure the flow circulation.

The closed-loop network for Program 7 is shown in Figure 4 in which the triplet on each arc is represented by (U_{ij}, L_{ij}, c_{ij}) for convenience. The visibility property of the network representation is clearly reflected in this diagram. At the termination of the OKA, the shortest path is found by tracing from supersource to supersink nodes over all arcs whose flow is equal to 1. Obviously, the decision variables on such a path are the optimal solution to the capacity expansion problem.

Numerical Examples

To demonstrate various solution procedures for the discrete time formulation, a small, yet representative, numerical ex-

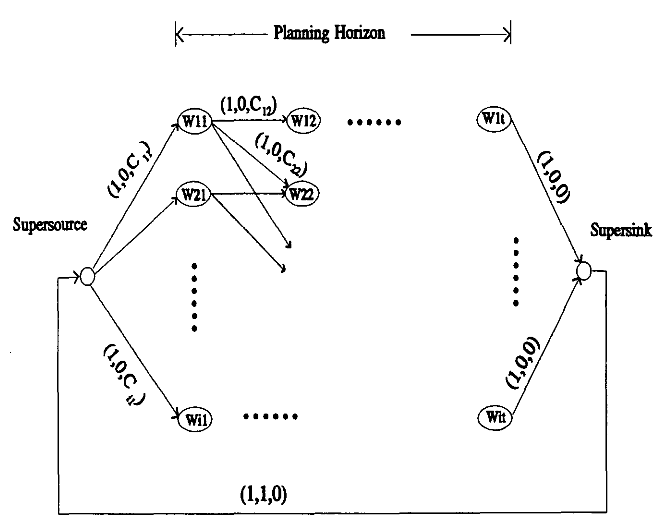


FIGURE 4 Out-of-kilter diagram for discrete time expansion problem.

ample is used. A system with three projects and five periods is considered. There are seven possible combinations of projects (i.e., seven alternatives) under the pairwise interactions assumption, and the increment set is identified in Table 1 for each alternative. It is assumed that the associated cost functions are available and monotonically increasing with respect to time. Quadratic polynomial functions seem appropriate for illustration. For computational convenience, the cost functions are sorted in ascending order of the constant terms and an index from 1 to 7 is assigned to each of them as given in Table 1. These cost functions represent some realistic considerations; for example, the costs of implementing several projects simultaneously are usually lower than implementing them individually, and single projects may be more costly than combinations. These cost functions are plotted in Figure 5.

According to Program 7 the costs of each alternative need to be calculated as the coefficients in the objective function. These values are computed from the area under each curve segment of the functions in Figure 5 and are listed in Table 1.

The OKA can now be applied to this illustrative case. According to the aforementioned procedures, an out-of-kilter

TABLE 1 Cost Functions for the Three-Project, Five-Period Example

Alt i	Project to be implemented	Increment Alternatives	Parameters* a	b	c
1	A only	1,3,5,7	2.5	4	4
2	B only	2,3,6,7	2.0	1	8
3	A & B only	3,7	0.45	5	10
4	C only	4,5,6,7	0.7	5	16
5	A & C only	5,7	1.0	0	18
6	B & C only	6,7	0.35	2	20
7	A & B & C	7	0.2	1.5	25

Alt i	Cost for Alt i in period t (C_{it}) 1	2	3	4	5	Total
1	6.83	15.8	29.8	48.8	72.8	174.03
2	9.17	14.2	23.2	36.2	53.2	135.97
3	12.7	18.6	25.3	33.1	41.7	131.4
4	18.7	25.1	32.9	42.1	52.7	171.5
5	18.3	20.3	24.3	30.3	38.3	131.5
6	21.1	23.8	27.2	31.3	36.1	139.5
7	25.8	27.7	30.0	32.7	35.8	152.0

*The cost function is assumed to be $f_i(t) = at^2+bt+c$

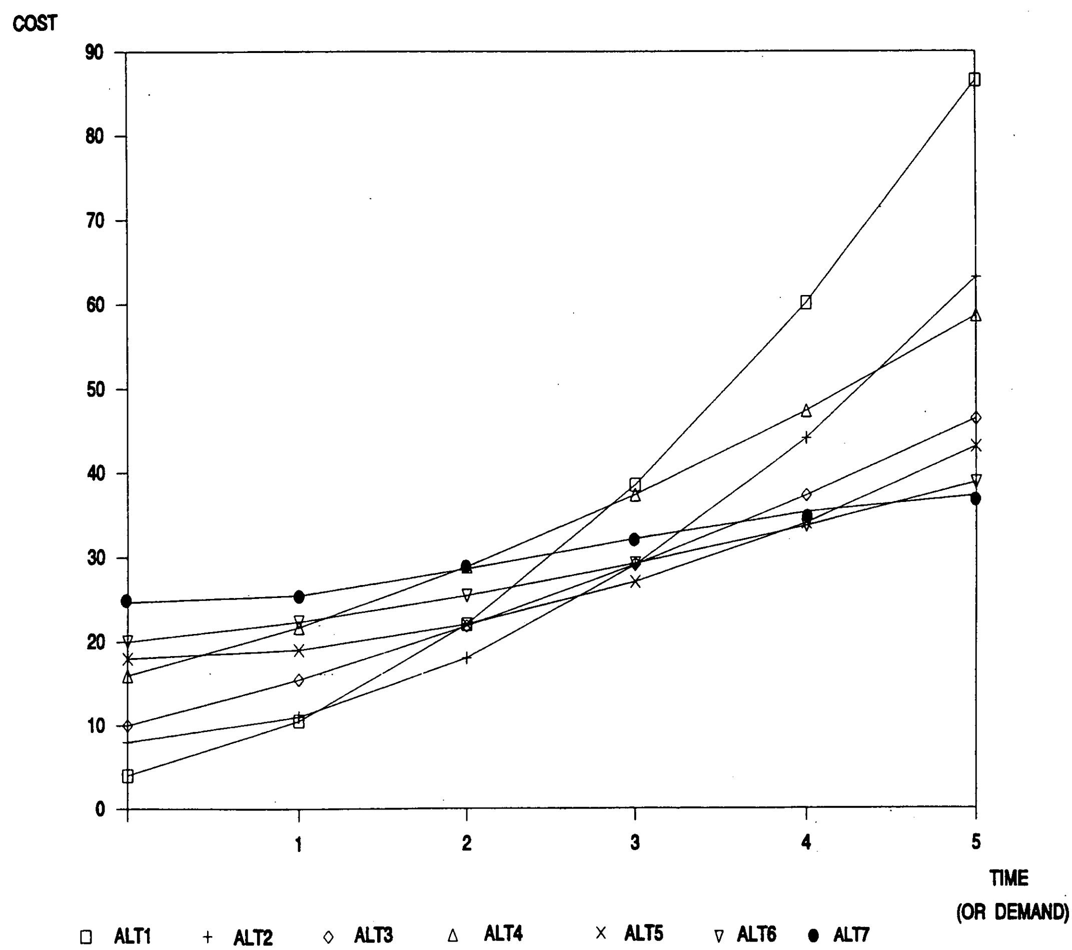


FIGURE 5 Cost functions for three-project, five-period illustrative case.

diagram is drawn in Figure 6. Table 1 provides information enough for this task, that is, nodes and arcs in the network and appropriate connections. The minimum cost path is needed from Node S to Node T in the out-of-kilter diagram, which is equivalent to directly solving Program 7 optimally. Four types of arcs are included in Figure 6, each associated with a vector representing the underlying integer program. The vectors for each arc indicate the allowable upper and lower bounds of flow and corresponding costs for every unit flow. The upper bound of flow is unity for all arcs, and the lower bound is 0 everywhere except for the arc between T and S to induce the flow through the network. Costs of flow on each arc depend on the origin and destination nodes, which correspond to the cost values in Table 1. The four types of arcs and associated vectors are listed in Table 2.

For an intermediate period t, arcs coming from alternative i (i.e., W_{it}) may go into alternative j for the next period (i.e., $W_{j,t+1}$) as long as alternative j is an element of increment set J_i. Because there should be one alternative for the first and last periods, Node S is linked to all variables for the first period and Node T is connected with all last-period variables. As a result, all variables in this diagram are directed to the next feasible decisions throughout the entire horizon. For instance, because $J_2 = \{2, 3, 6, 7\}$, W_{21} can be linked only with W_{22}, W_{32}, W_{62}, and W_{72} in the second period. Note that the structures between columns of W_{it}'s are identical, which is indeed one advantage of such representation, namely, the connections between W_{i2} and W_{i3} are the same as those between W_{i1} and W_{i2}, and so on. This particular structure ensures the project continuity constraint at any period and implicitly incorporates all feasible solutions and corresponding costs. Because of its simplicity, this diagram and associated information can be easily translated into a computer program and then the OKA can be used.

When budget constraints are included, a number of links and nodes in the diagram may be eliminated, most likely the multiproject alternatives that require higher costs. The resulting diagram could be reduced, thus increasing the efficiency of the OKA.

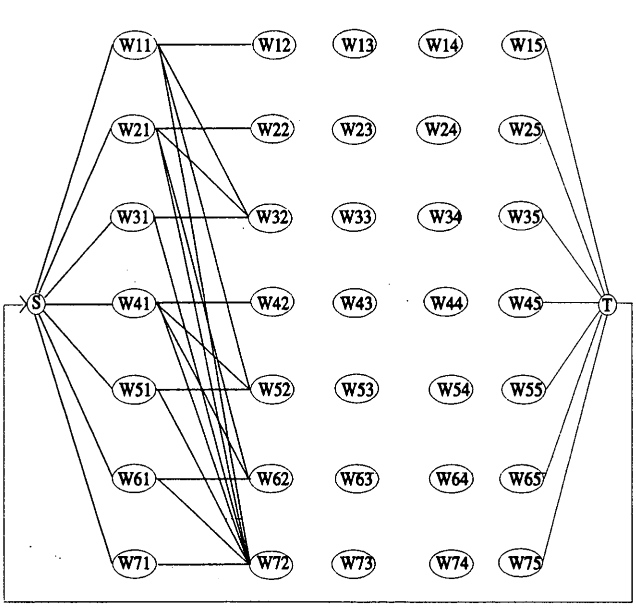


FIGURE 6 Out-of-kilter diagram for the three-project, five-period case.

TABLE 2 Arc Characteristics in Out-of-Kilter Diagram

Arc type	Vector (U, L, c)*
S --> W_{i1}	(1, 0, C_{i1})
W_{i5} --> T	(1, 0, 0)
T --> S	(1, 1, 0)
$W_{i,t}$ --> $W_{j,t+1}$	(1, 0, $C_{j,t+1}$)

*U, L, and c are upper and lower bounds of flow, and cost of unit flow

The optimal solution is $W_{11} = W_{12} = W_{53} = W_{54} = W_{75} = 1$, other W_{it}'s equal 0, and a total cost of 113.03 units. According to Table 1, the optimal investment program is to start Project A immediately, add Project C at the beginning of the third period, and finally add Project B for the last period. The solution for project selection, sequencing, and scheduling may be represented simply by the following expression:

$$A \rightarrow A \rightarrow A + C \rightarrow A + C \rightarrow A + B + C$$

To demonstrate the relative advantage of combining the integer program with the OKA for the project scheduling problem, two much larger examples are generated arbitrarily. The information on these examples is summarized in Table 3, and some representative cost functions are plotted in Figure 7. The curves in Figure 7 represent various subsets of alternatives, each with a different number of projects to be implemented simultaneously. It is clear that these two examples are quite practical. Besides, although the costs all increase over time, the functions in Figure 7*a* have increasing first derivatives, whereas those in Figure 7*b* yield decreasing first derivatives. This may reflect the flexibility of the proposed approach.

The LINDO program is also used to solve the above cases. For the three-project, five-period case, only eight iterations are required to reach the optimal solution. The computational efficiency has been greatly improved, compared with 3,632 iterations for the previous work (*12*). For the two larger examples, the computation times were found to be comparable with those using the OKA. Although such a situation implies that LINDO performs as well as the OKA, it also confirms the effectiveness of the alternative-based formulation. One should note, however, that LINDO is a commercial software and is possibly an order of magnitude more efficient than the OKA code. Therefore, although LINDO can readily be used to solve the resulting linear programs from converting the original nonlinear programs, it may still run into long computation time when problems become larger. On the other hand, an efficient code for the OKA seems especially suitable for the discrete expansion problem and is conceivably more efficient than LINDO for relatively large problems.

CONCLUSIONS

The capital budgeting problem for interrelated capacity expansion projects is discussed in this paper. An integer program

TABLE 3 Information on Two Larger Problems

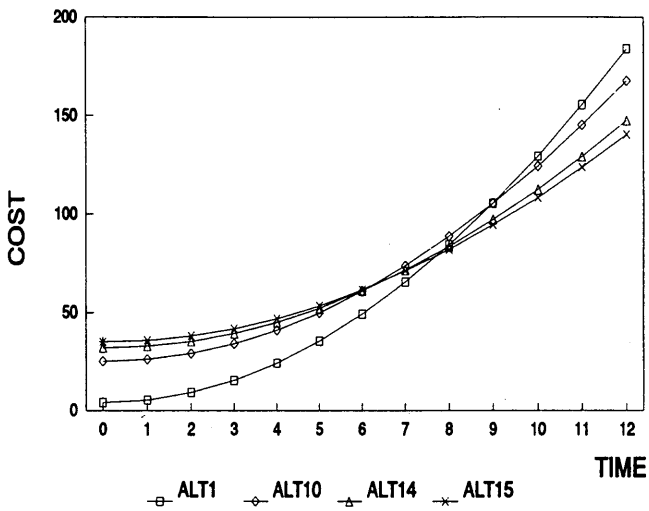
Example	# Projects	# Alts	# Periods	# Decision Variables	Cost Function	CPU* (seconds)
1	4	15	12	180	$at^2 + c$	29
2	5	31	12	372	$at^{1/3} + c$	172

*Data obtained in IBM PS/2 Model 70 with 20 MHz speed

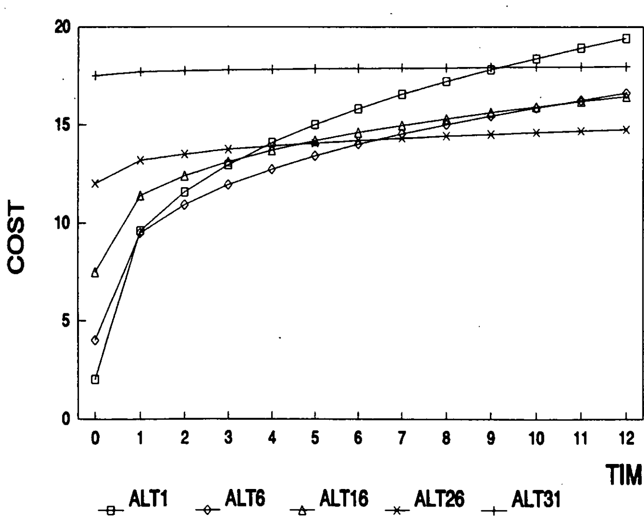
is proposed and two distinct procedures are used to solve the problem optimally. The interactions among projects are specified before programming to enhance the model efficiency. The alternative-based approach treats alternatives as mutually exclusive decision variables. The project continuity constraint is ensured by defining the incremental relationships among alternatives, which is found to be an underlying advantage of the proposed model.

A network representation procedure is the primary solution method presented. The model is illustrated to be easily translated into a directed network with nice properties. The out-of-kilter algorithm is applied to this network problem, and the results of several examples reveal the computational efficiency of such an approach. Although LINDO yields similar performance for these illustrative cases, relatively large problems could be formulated and solved more efficiently with this network optimization method. Besides, the proposed procedure generates exactly optimal solutions instead of heuristic solutions.

The proposed discrete time alternative-based formulation is likely to result in large integer programming problems as the number of projects increases. However, because the proposed solution procedure is very efficient, it is capable of handling problems with thousands of decision variables. Furthermore, because in the real world it is likely that there exists a limited number of distinct projects with pairwise interactions, the number of alternatives would be manageable. Nevertheless, decomposing the system of interest by any reasonable criteria should be considered whenever possible to reduce the complexity and computational efforts. With small modifications, such as given benefit rather than cost functions, the proposed model could be used when total benefit is the primary concern. Further work may be done that incorporates more precise project interactions with the corresponding cost/benefit functions, and other solution procedures.



(a) 4-project, 12-period Example



(b) 5-project, 12-period Example

FIGURE 7 Representative cost functions for larger problems.

REFERENCES

1. D. Erlenkotter. Sequencing of Interdependent Hydroelectric Projects. *Water Resources Research*, Vol. 9, No. 1, 1973.
2. T. E. Gear and G. C. Cowie. A Note on Modeling Project Interdependence in Research and Development. *Decision Science*, Vol. 11, 1980.
3. H. Luss. Operations Research and Capacity Expansion Problems: A Survey. *Operations Research*, Vol. 30, No. 5, 1982.
4. A. F. Czajkowski and S. Jones. Selecting Interrelated R&D Projects in Space Technology Planning. *IEEE Transactions on Engineering Management*, Vol. 33, No. 1, 1986.
5. L. F. A. M. Gomes. Modelling Interdependencies Among Urban Transportation System Alternatives Within a Multicriteria Ranking Framework. *Journal of Advanced Transportation*, Vol. 24, No. 1, 1990.
6. V. Srinivasan and Y. H. Kim. Evaluating Interrelated Capital Projects: An Alternative Framework. *The Engineering Economist*, Vol. 33, No. 1, 1987.
7. R. E. Steuer and F. W. Harris. Intra-Set Point Generation and Filtering in Decision and Criterion Space. *Computer and Operations Research*, Vol. 7, 1980.
8. J. N. Morse. Reducing the Size of the Nondominated Set: Pruning by Clustering. *Computer and Operations Research*, Vol. 7, 1980.
9. B. N. Janson et al. Network Design Programming of U.S. Highway Improvements. ASCE *Journal of Transportation Engineering*, Vol. 117, No. 4, 1991.
10. V. Akileswaran et al. Complexity of the Project Sequencing Problem. *Operations Research*, Vol. 31, No. 4, 1983.
11. D. Martinelli. *Investment Planning of Interdependent Waterway*

Improvement Projects. Ph.D. dissertation. University of Maryland, College Park, 1991.
12. C. H. Wei. *Optimization of Capital Budgeting*. Working Paper ENCE 681. University of Maryland, College Park, 1991.
13. S. M. Shafer and S. J. Mantel, Jr. A Decision Support System for the Project Termination Decision: A Spreadsheet Approach. *Project Management Journal*, Vol. 20, No. 2, 1989.
14. J. F. Bard et al. An Interactive Approach to R&D Project Selection and Termination. *IEEE Transactions on Engineering Management*, Vol. 35, No. 3, 1988.
15. D. T. Phillips and A. Garcis-Diaz. *Fundamentals of Network Analysis*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.

Publication of this paper sponsored by Committee on Transportation Programming, Planning, and Systems Evaluation.