

A NETWORK-RELATED NUCLEAR POWER PLANT MODEL WITH AN INTELLIGENT BRANCH-AND-BOUND SOLUTION APPROACH *

Fred GLOVER¹, Darwin KLINGMAN² and Nancy V. PHILLIPS³

¹ *U.S. West Chair in Systems Science, School of Business, University of Colorado, Boulder, CO 80309-0419, U.S.A.*

² *Hugh Roy Cullen Centennial Chairholder in Business Administration; Director of the Center for Business Decision Analysis; and Professor of Management Science and Information Systems, Graduate School of Business, and Computer Sciences, College of Natural Sciences, The University of Texas at Austin, Austin, TX 78712, U.S.A.*

³ *Assistant Director of the Center for Business Decision Analysis; and Associate Professor of Management Science and Information Systems, Graduate School of Business, The University of Texas at Austin, Austin, TX 78712, U.S.A.*

Abstract

This paper presents a network model with discrete requirements for a nuclear power plant. The model determines the batch size and timing for nuclear unit refueling and how much energy should be produced by nuclear and non-nuclear units for each time period to satisfy forecasted demand with minimum total operating costs over the planning horizon. Efficient modeling and solution strategies are developed which constitute a merger of operations research and artificial intelligence. A branch-and-bound solution approach is combined with a pattern recognition component, involving non-parametric discrimination analyses, to select branching variables and directions. By coupling this approach with network optimization techniques to exploit the underlying network structure of the problem, substantial improvements are obtained both in solution quality and solution efficiency.

1.0. Introduction

Management science techniques have been applied to all aspects of nuclear power plant operation. In the area of system scheduling and planning, operations research techniques used include simulation (Turnage and Prince [22]), linear programming (Deaton [2] and Rhodes [21]), mixed integer programming (Kazemersky [16]), critical path method (Catalano [1]), expert systems (Nuclear Services [20]), and decision analysis (Hamilton and Bingham [13]).

* This research was supported in part by the Center for Business Decision Analysis, the Hugh Roy Cullen Centennial Chair in Business Administration, and the Office of Naval Research under contract N00014-87-K-0190. Reproduction in whole or in part is permitted for any purpose of the U.S. Government. CENTER FOR BUSINESS DECISION ANALYSIS Darwin Klingman, Director The University of Texas at Austin Austin, TX 78712, U.S.A.

All of the foregoing scheduling models, except for the one by Kazemersky [16], assume that the refueling dates are known a priori. In this paper, we present a model which is mathematically equivalent to Kazemersky's but which is much easier for management to understand and which, based on preliminary empirical testing, appears to be more tractable from a solution perspective.

Scheduling future energy production for fossil, nuclear, and hydro units and nuclear refueling for a planning horizon of several years is a complex problem. Generally, a utility attempts to select the schedule which minimizes the total system present-value operating cost to reliably service its forecasted demand load.

Nuclear and fossil fuel costs are the major components of a utility system's operating cost. Nuclear fuel is a highly processed, manufactured product which requires substantial lead time and financial investment. Fuel normally remains in-core (in the reactor) from two to four years, and when removed requires additional time and financial investment to process and recover useful products. Furthermore, the past operation of the nuclear unit influences the present energy capability of the unit since only a portion (batch) of the fuel in a reactor is replaced at each refueling, and a unit cannot produce energy while being refueled. Consequently, the economic trade-offs between energy production (nuclear and non-nuclear) and refueling decisions must be examined simultaneously. The effects of past, present, and future operation on the cost of a fuel batch, and the lifetime of a batch generally require a planning horizon of five to seven years.

For this application, we developed a NETFORM (i.e., a network flow related model). This NETFORM is a network model with accompanying discrete flow requirements to identify batch sizes and timing for nuclear refueling and to determine how much energy should be produced by each unit (nuclear and non-nuclear) for each time period to satisfy the forecasted demand. The objective is to minimize the total system present-value operating costs over the planning horizon of the model. Because the problem has a large number of discrete requirements, it requires an effective blend of modeling and solution strategies.

Kazemersky [15] developed a mixed-integer programming model for this problem. His model formulation requires over twenty pages to present and is difficult to understand from a managerial perspective. Our equivalent NETFORM can be presented in a few pages and understood easily by management. The strategy we undertook for solving this problem constitutes a merger of operations research and artificial intelligence, based on a procedure for learning the best way to search the solution space. A branch-and-bound solution approach is combined with a pattern recognition component, involving non-parametric discrimination analyses, to select branching variables and directions. Using these methods, together with network optimization techniques to exploit the underlying network structure of the problem, we were able to substantially reduce solution times and improve solution quality. Four versions of the refueling problem were solved using data supplied by the Tennessee Valley Authority (TVA). Solution times for the first three versions, using MPSX on an IBM 4381, were half an hour to two

hours. By contrast, the same problems were easily solved in 1 second to 20 minutes using our NETFORM and a specialized branch-and-bound solution approach. The fourth version, which involved 172 constraints, 126 discrete requirements, and 462 continuous variables (arcs) was by far the most difficult. The original mixed-integer formulation was run for seven hours on an IBM 4381, again using MPSX, and was then taken off the machine to avoid further computer run costs. The best (minimum cost) solution obtained had an objective function value of \$136,173,440. With a 30 minute time limit imposed on the NETFORM procedure, a solution was obtained that was more than \$10,000,000 cheaper, with an objective function value of \$125,174,727. This application demonstrates that an innovative modeling/solution method that blends appropriate elements of operations research and artificial intelligent can obtain solutions to problems that are too complex to be solved optimally (within practical time limits) by standard approaches.

2.0. Model development

Several assumptions were made about the utility system, following the guidelines reported in Kazemersky [15,16]. First, the length of the time periods are fixed to the average length of a refueling outage. Based on historical refueling data, the length of each time period was chosen as two months. Second, we assume that nuclear maintenance occurs during refueling, fossil maintenance occurs on the capacity which is not producing energy, and there are no forced outages.

A network model with discrete requirements for the energy scheduling and refueling problem for a utility system with three nuclear units and two non-nuclear units is shown in fig. 1. (For a review of network terminology see for example Glover et al. [6,7], Glover and Klingman [9], Jensen and Barnes [14], and Kennington and Helgason [18].) The node at the top of the figure, labeled R , represents refueling. It is connected by an arc to each of a set of nodes labeled P_{kt} which represents nuclear unit k in time period t . These arcs have lower and upper bounds rl and ru , shown in parentheses for the arc into node P_{kn} , which represent the minimum and maximum energy that can be refueled into each unit during each time period. (The other components of the bound notation, which represent the discrete requirements, will be discussed later.) The costs on these arcs, rc , shown in rectangles, represent the present value procurement cost of the fuel batch, expressed as a per unit energy cost. The supply to the refueling node, indicated by the r in the triangle pointing toward the node, is the sum of all possible refueling activities over the planning horizon and is non-binding at optimality.

Each nuclear unit node in the first time period, P_{k1} , has supply a_k which represents the initial state of the in-core fuel (energy level) at each unit. Likewise,

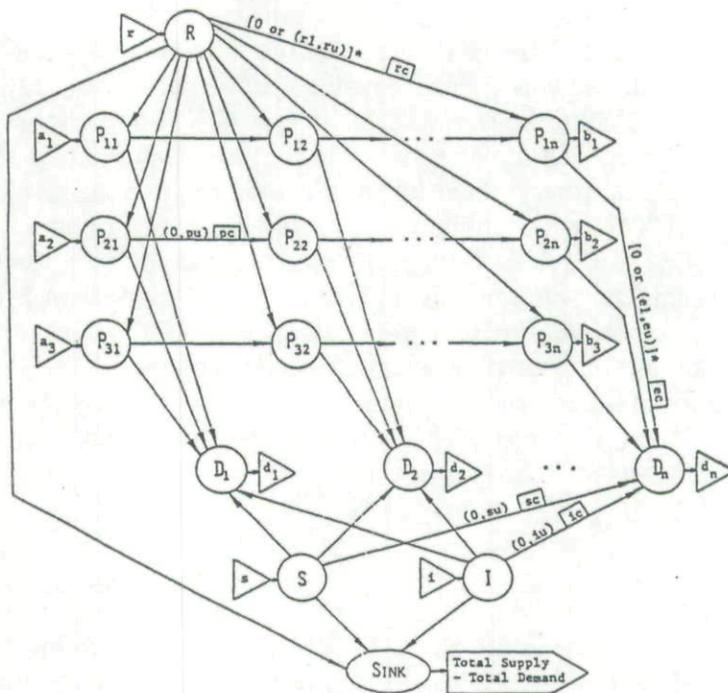


Fig. 1. Refueling model.

each nuclear unit in the last time period, node P_{kn} , has a demand b_k , shown in the triangle pointing away from the node, which represents the desired ending state of the in-core fuel (the "equilibrium" energy level). The nuclear unit nodes are interconnected by arcs which represent the inventory of nuclear energy from one time period to the next. These arcs have lower bounds equal to zero and upper bounds, denoted pu , which represent the maximum energy inventory allowed (shown for the inventory arc from node P_{21} to node P_{22}). The costs on these arcs, pc , represent the present value of the interest cost applicable to the uranium fuel in-core, expressed as a per unit energy cost. The nuclear unit nodes in time period t are each connected by an arc to a node labeled D_t , which represents the utility system customers who have a total energy demand of d_t , during time period t . The bounds on these arcs el and eu (shown for the arc out of node P_{1n}) represent the minimum and maximum energy that each nuclear unit can produce during each time period. (The other components of the bound notation, which represent the discrete requirements, will be discussed later.) The costs on the energy production arcs, etc, are the present value of the operating and maintenance cost of the nuclear unit (including cost of refueling but not fuel costs) expressed as a per unit energy cost.

The two nodes labeled S and I near the bottom of fig. 1 represent the non-nuclear units of the utility system and energy obtained from other utilities.

The node labeled S represents the non-nuclear energy balance of the utility system with supply s equal to the sum of all possible non-nuclear energy production activities over the planning horizon. The non-nuclear units node is connected by an arc to each customer demand node. The arcs have a lower bound of zero and an upper bound su , the maximum amount of energy that the non-nuclear units can produce during each time period (shown for the arc into node D_n). The costs, sc , are a weighted combination of all the present value non-nuclear costs expressed as a per unit energy cost. The node labeled I represents energy interchange arrangements with other utilities. It has a supply and arcs with bounds and costs analogous to the non-nuclear unit node. The sink node at the bottom of fig. 1 balances total supply and total demand for the network. The optimal solution to the model indicates energy and refueling requirements for the nuclear and non-nuclear units in each time period and cycle lengths (refueling dates).

We note that the realism of this portion of the model can be enhanced by adding multipliers on the arcs to represent the half-life of the uranium being used and/or transmission line losses, thereby producing a generalized network-related model. However, since previous models did not include these aspects, and no data concerning them was available, we restricted our attention to a pure network-related model.

The discrete requirements of the model derive from the technology of nuclear energy production. The bound notation $[0 \text{ or } (l, u)]^*$ on the refueling arc into and demand arc out of each nuclear unit node (shown for node P_{1n} in fig. 1) indicate that both of these arcs cannot have flow simultaneously. That is, a nuclear unit is prohibited from refueling and producing energy during the same time period. This type of condition is not part of the standard network framework and may be handled using zero-one variables. More formally, a mixed-integer programming model results by expanding the network formulation to include constraints of the form

$$E_{kt} \leq eu_{kt}(X_{kt}) \quad k = 1, 2, 3 \quad t = 1, 2, \dots, n$$

$$E_{kt} \geq el_{kt}(X_{kt}) \quad k = 1, 2, 3 \quad t = 1, 2, \dots, n$$

$$R_{kt} \leq ru_{kt}(1 - X_{kt}) \quad k = 1, 2, 3 \quad t = 1, 2, \dots, n$$

$$R_{kt} \geq rl_{kt}(1 - X_{kt}) \quad k = 1, 2, 3 \quad t = 1, 2, \dots, n$$

where E_{kt} denotes the decision variable corresponding to the energy produced at nuclear unit k during time period t , R_{kt} denotes the decision variable corresponding to the energy refueled at nuclear unit k during time period t , and

$$X_{kt} = \begin{cases} 1 & \text{if nuclear unit } k \text{ is operating during time period } t \\ 0 & \text{if nuclear unit } k \text{ is refueling during time period } t. \end{cases}$$

The X_{kt} variables have zero objective function coefficients since the refueling costs are reflected in the costs ec and rc .

A typical integer programming approach to solve this problem would be to relax the integer restriction on the X_{kt} variables. The resulting relaxation is a linear programming (LP) problem with $16n + 4$ constraints and $14n$ variables. This LP problem contains a large embedded network consisting of $4n + 4$ constraints (nodes) and $11n$ variables (arcs). This embedded network structure can be exploited using a basis partitioning LP algorithm (Glover and Klingman [10], Kennington and Helgason [18], and Graves and McBride [12]). For example, the simplex SON algorithm (Glover and Klingman [10]) has proven to be highly efficient for this class of LP problems. However, LP codes are numerically less stable than network codes and also less portable. (Generally, LP codes have a factor of ten times more lines of code than network flow codes.) To avoid these limitations and yet still take advantage of the embedded network structure, we replace the integer programming constraints for the discrete requirements by the following nonlinear conditions:

$$\begin{aligned}
 E_{kt}R_{kt} &= 0 & k &= 1, 2, 3 & t &= 1, 2, \dots, n \\
 \text{if } R_{kt} > 0 & \text{ then } rl_{kt} \leq R_{kt} \leq ru_{kt} & k &= 1, 2, 3 & t &= 1, 2, \dots, n \\
 \text{if } E_{kt} > 0 & \text{ then } el_{kt} \leq E_{kt} \leq eu_k & k &= 1, 2, 3 & t &= 1, 2, \dots, n.
 \end{aligned}$$

The relaxation employed in our solution approach is to ignore the constraints $E_{kt}R_{kt} = 0$ and set $rl_{kt} = 0$ and $el_{kt} = 0$. These nonlinear constraints have certain advantages over the mixed-integer constraints. The nonlinear model does not introduce any new variables to the problem. The nonlinear relaxation results in creating smaller subproblems, which are network flow problems. However, the nonlinear relaxation is weaker than the standard LP relaxation, the one with the X_{kt} variables, and thus our solution method requires a stronger (more intelligent) search technique.

Whether these advantages can be translated into practical gains depends on the ability to develop a procedure to exploit the nonlinear constraints effectively within the network framework. The next section describes our approach for accomplishing this.

3.0. Solution strategy

The integration of artificial intelligence and operations research in our approach is based on the principle of using structure that is known in advance, and discovering structure that is initially unknown, to produce a strategy capable of handling the complex combinatorial elements of the problem. The operations research component of our approach consists of a branch-and-bound strategy tailored to solve network subproblems efficiently by invoking the lower bounds rl_{kt} and el_{kt} and restarting from the old optimal basis. The artificial intelligence component consists of a pattern recognition and learning strategy, involving non-parametric discrimination analysis to determine good rules for choosing

variables on which to branch. The general concept of such a strategy for solving integer programming problems, which we refer to as *target analysis*, has been described in (Glover [4]).

Our present use of target analysis takes the following form. When the nuclear refueling NETFORM of section 2 is solved in the relaxed nonlinear form as a network, the solution violates the discrete requirements whenever $E_{kt}R_{kt} > 0$ for some k and t . Thus, for each node (branching point) in the branch-and-bound tree, the binary branching alternatives are $R_{kt} = 0$ or $E_{kt} = 0$ for all kt pairs which are candidates for branching at the node. Let $s(j)$ denote the kt pair associated with the branching alternative j . (The index j receives a different range of values for different nodes, and hence identifies both a node and a candidate pair of branching alternatives at the node.) Select a vector v_j of parameters and let $F_i(v_j)$, $i \in M = (1, \dots, m)$ denote a set of evaluation functions created to assess the relative attractiveness of the two possibilities for the binary branch j at a particular node. Our ultimate goal is to determine a composite evaluation function and a threshold function value which will be used to select branching variables and directions at each node in the branch-and-bound tree.

Because of the large number of discrete requirements in this application, and the desire to test the target analysis approach with information easily and quickly obtainable from the relaxed network solution, we restrict attention to the values of the two variables $R_{s(j)}$ and $E_{s(j)}$ to make up the components of the vectors v_j . To create evaluation functions based on these parameters, it is useful first to consider ways of characterizing the degree to which a particular discrete requirement is violated by a given solution. The following measures will be 0 if the requirement is satisfied and will be positive otherwise. Hence their magnitude gives a measure of the degree of violation.

- (1) $R_{kt}E_{kt}$
- (2) $R_{kt} + E_{kt} - |R_{kt} - E_{kt}|$
- (3) $R_{kt}^2 + E_{kt}^2 - |R_{kt}^2 - E_{kt}^2|$
- (4) $\text{Min}[(R_{kt}/E_{kt}), (E_{kt}/R_{kt})]$
- (5) $\text{Min}[(R_{kt}/E_{kt})^2, (E_{kt}/R_{kt})^2]$

The cases where denominators vanish in (4) and (5) are interpreted in the natural fashion for this context; that is, the ratio a/b is interpreted as infinity if $b = 0$, allowing for the exception $0/0 = 0$. (In all cases, the min is 0 whenever one of the numerators is 0.)

It is also possible to raise selected terms to higher powers to get additional measures. (In one variation, for example, the measure (2) itself could be squared, instead of squaring its component terms as in (3). The measure $(R_{kt} + E_{kt})^2 - (R_{kt} - E_{kt})^2$, however, is equivalent to (1).) For simplicity, we elected to use only the measures shown.

A natural branching alternative for a branch-and-bound framework is to choose one of R_{kt} and E_{kt} to force to 0 when both are positive. It might be supposed that the "normal" choice would be to select the smaller of these values (or, in a broader context, the one that entails a smaller penalty calculation). The probable validity of such a choice, however, can also depend on the relative magnitudes of R_{kt} and E_{kt} and on the relationship of these magnitudes to those associated with other candidate kt pairs.

Specifically, the evaluation functions we selected as potentially relevant for target analysis consisted of the measures (1)–(5) themselves, together with the ratios of (1) and (2) to the values R_{kt} and E_{kt} , and the ratio of (3) to the values R_{kt}^2 and E_{kt}^2 . The ratios provide a natural means for identifying which of R_{kt} and E_{kt} should "more likely" receive a 0 value (in inverse relation to the magnitude of the ratio), but this indication may be misleading, and hence arises the need to incorporate these values into a discrimination analysis model. For this purpose, to differentiate the role of the measures (1)–(5) when not used as numerators for ratios, we reverse the sign of the measures when $R_{kt} < E_{kt}$. We also used one additional evaluation function consisting of $(R_{kt} - E_{kt}) / (R_{kt} + E_{kt})$.

Note that the ratios based on dividing the measure (1) by R_{kt} and E_{kt} yield R_{kt} and E_{kt} themselves, in reverse order, as evaluation functions. Also, measure (2) may be seen equivalent to the value $2 \min(R_{kt}, E_{kt})$; hence the ratios based on this value will be 2 and $2 \min[(R_{kt}/E_{kt}), (E_{kt}/R_{kt})]$. It is important to differentiate these ratios according to which one results by dividing by R_{kt} and which one results by dividing by E_{kt} . (Similar remarks apply to the ratios obtained from measure (3).)

Since the magnitude of the evaluation functions gives a measure of the degree of violation of the discrete constraints, an example rule to choose a branching direction for branching alternative j using evaluation function i is to determine a value b_i such that the direction of branching is given by:

$$R_{s(j)} = 0 \text{ if } F_i(R_{s(j)}, E_{s(j)}) \leq b_i$$

$$E_{s(j)} = 0 \text{ otherwise.}$$

Within the branch-and-bound framework we must go beyond the choice of branching direction, however, and choose a specific branch j as a best candidate at a node for applying the preceding rule. In particular, a choice suggested by the rule itself is to pick a branch q such that

$$|F_i(R_{s(q)}, E_{s(q)}) - b_i| = \max |F_i(R_{s(j)}, E_{s(j)}) - b_i|$$

where the max is over those j which are candidates for branching at the node.

To complete the foregoing specification, it is necessary to identify a specific index i , and hence a specific evaluation function $F_i(R_{s(j)}, E_{s(j)})$, for which the choice should be applied. Among the possibilities for doing this, it seems appropriate to allow the chosen function to result by combining the information from the evaluation functions for the other indexes $i \in M$. In (Glover and McMil-

lan [11]), for example, such evaluators were allowed to "vote" on different alternatives. Another possibility is to create a decision tree that organizes the $F_i(R_{s(j)}, E_{s(j)})$ functions hierarchically to yield a conditional partitioning of possibilities. In our present development, to provide a convenient framework for the learning feature of target analysis, we elected to use discrimination analysis to generate a function $F(R_{s(j)}, E_{s(j)})$ which serves as a "best proxy" for the candidate functions $F_i(R_{s(j)}, E_{s(j)})$, yielding at the same time an associated value b as a best proxy for the b_i values, so that a decision rule of the foregoing type can be applied.

3.1. DISCRIMINATION ANALYSIS FORMULATION

To understand this approach, suppose it is possible to consult an oracle that provides knowledge of optimal values $R_{s(j)}^*$, $E_{s(j)}^*$ for every $R_{s(j)}$ and $E_{s(j)}$ at each node. With each branch j we may associate the point $A_j = (F_1(R_{s(j)}, E_{s(j)}), E_{s(j)}, F_2(R_{s(j)}, E_{s(j)}), \dots, F_m(R_{s(j)}, E_{s(j)}))$ in m -space. Considering the set of all branches at all nodes and their associated points A_j , it then becomes natural to partition these points into two groups, with index sets R_0 and E_0 , based on the $R_{s(j)}^*$, $E_{s(j)}^*$ values; that is,

$$R_0 = \{j: R_{s(j)}^* = 0\}$$

$$E_0 = \{j: E_{s(j)}^* = 0\}$$

Within this setting, we seek a rule, based only on knowing the coordinates of points A_j , for assigning each A_j to the group R_0 or E_0 which minimizes the number of points incorrectly assigned.

As we will subsequently demonstrate, however, some points are more important than others to classify correctly according to their membership in R_0 or E_0 . Moreover, it is also more important to assure correct classification of "strongly assigned" points than it is to assure correct classification of a large number of points. Motivated by these considerations, we make use of the linear programming model for discrimination analysis developed in Glover et al. [8] and Glover [5]. The basis of this approach is to find a hyperplane that separates the points of R_0 and E_0 , as nearly as possible, where points can be differentially penalized for lying on the wrong side of the hyperplane, and differentially rewarded for lying on the right side, expressing these penalties as a function of the L_1 norm ("absolute value" or "city block") distances from the hyperplane.

Adapted to the present context, this model may be expressed as follows. Let J be the union of R_0 and E_0 ; that is, J is the index set for all points A_j . We seek a weighting vector x and a scalar b in order to:

$$\text{Minimize} \quad \sum_{j \in J} h_j \alpha_j - \sum_{j \in J} k_j \beta_j \quad (1)$$

$$\text{subject to } A_j x - \alpha_j + \beta_j = b; \quad j \in R_0 \quad (2)$$

$$A_j x + \alpha_j - \beta_j = b; \quad j \in E_0 \quad (3)$$

$$\#E_0 \sum_{j \in R_0} (\beta_j - \alpha_j) + \#R_0 \sum_{j \in E_0} (\beta_j - \alpha_j) = 1 \quad (4)$$

$$x, b \text{ unrestricted} \quad (5)$$

$$\alpha_j \text{ and } \beta_j \geq 0 \quad j \in J \quad (6)$$

where $\#S$ denotes the cardinality of the set S .

The α_j and β_j variables respectively represent "external" and "internal" deviations which measure how far the points A_j lie outside or inside the half-spaces intended to contain them. (These half-spaces are $A_j x \leq b$ for j in R_0 and $A_j x > b$ for j in E_0 .) The objective function coefficients h_j and k_j are chosen so that $h_j > k_j$ (discussed in more detail in section 3.3). Equation (4) is a *normalization constraint*, whose form is important for the effectiveness of the model (see, e.g., Glover [5]).

The solution to the linear problem (1) – (6) yields a vector x and scalar b that provides the desired "weighted average" function $F(R_{s(j)}, E_{s(j)}) = A_j x$, and scalar b , for target analysis. The issue of choosing a branching variable and a direction of branching is resolved by finding the point A_j that falls most deeply in one of the two half-spaces defined by the LP solution, and then branching in the direction $R_{s(j)} = 0$ if the half-space is associated with R_0 and in the direction $E_{s(j)} = 0$ if the half-space is associated with E_0 . Thus, in particular, we choose a branching alternative q such that

$$|A_q x - b| = \max |A_j x - b|$$

where the max is over all j that are candidates for branching at a particular node. We then choose $R_{s(q)} = 0$ or $E_{s(q)} = 0$ according to whether $A_q x \leq b$ or $A_q x > b$.

3.2. EMBEDDING THE DISCRIMINATION MODEL IN TARGET ANALYSIS

The strategy of target analysis replaces the oracle for determining the optimal $R_{s(j)}$ and $E_{s(j)}$ values in this approach by a series of *learning trials*. Each of these trials selects a problem whose structure is representative of the class under consideration, and solves this problem to optimality. (Since the problems of interest here are quite large, we selected smaller problems for the learning trials, so that the solution effort was not overwhelming.) Once an optimal solution is known, the approach then goes back and re-solves the problem, this time selecting an optimal branching direction at each node. There is, of course, more than one way to carry out this process, depending on which variable is selected as the branching variable at each step.

Because the sequence of branches chosen in the "re-resolution" phase is compatible with an optimal solution, the $R_{s(j)}^*$ and $E_{s(j)}^*$ values are known at each node,

and we can identify the sets R_0 and E_0 . Upon generating the points A_j from the information directly available at each node, we may therefore create and solve the linear program (1)–(6) for discrimination analysis, as previously described. The resulting half-spaces are then used to select branching variables and directions in the application of the branch-and-bound solution approach to the larger problem.

It should be observed that data from different nodes can yield different points A_j associated with the same variable of a given problem. This poses no difficulty, since every branching alternative encountered in the learning trials is given a different index j , regardless of whether the same variable is associated with different branches.

3.3. CHOOSING OBJECTIVE FUNCTION COEFFICIENTS FOR THE DISCRIMINATION MODEL

The final element required to implement the target analysis procedure is to choose values for the coefficients h_j and k_j of the LP discrimination model. Note that a branch-and-bound procedure can find its way directly to an optimal solution, under appropriate conditions, without classifying all the A_j points correctly according to their membership in R_0 or E_0 . In particular, since only one branch will be chosen at any node, it is necessary only to make sure that the point for this chosen branch is correctly classified. Which branch should this be?

The answer we give is based on the supposition that it is important to select an *influential* branch (see, e.g., Glover [3]). Loosely speaking, we conceive a branch q to be influential if the alternatives $R_{s(q)} = 0$ and $E_{s(q)} = 0$ (or the preferred member of these alternatives) will cause the evaluators $F_i(R_{s(j)}, E_{s(j)})$ to be appreciably different at the descendants of the current node than they are at the node itself. Thus, a branch is influential if it has the power to change the information available for analysis after making a branch step.

This notation is based on the assumption that a change in information content signals a useful increase in this content—an assumption that is generally true upon executing a branching step in a branch-and-bound method. Thus, more precisely, the notion of an influential branch is linked to the expectation of an increase in exploitable knowledge.

The intuitive basis for this characterization is the following empirical observation. When an integer programming problem is solved by branch and bound, one common approach is to use the simplex method to solve the LP relaxation of the integer problem at each node. A branching option in this approach (not often used) is to choose a variable that is nonbasic and branch in its zero direction. This is evidently the “preferred” direction since the variable receives a zero value in the LP solution.

Such a branching option may be expected to be poor, because taking the preferred branch direction yields no new information at the next node; that is, the same LP solution remains optimal. The advantage of obtaining progressively

more limiting information at deeper levels of the branch-and-bound tree, therefore, is sacrificed. Experience confirms expectation, disclosing that such a choice rule can sometimes require effort approximating that of total enumeration.

A better choice evidently is to branch on a variable whose value is not at, or extremely close to, an integer value in the LP solution. A quick and dirty rule for zero-one variables found reasonably good in Klingman et al. [19] is to branch on a variable whose value is closest to 0.3 or 0.7, choosing the alternative that moves to the nearest integer. Values closer to 0.5, while potentially associated with more influential branches, are less effective (in the absence of further information), due to the increased risk that branching to the nearest integer is an improper choice. This choice rule is appropriate for problems that are quite large or, in general, where branching information is expensive to generate.

The issue in the present context, therefore, is to choose values for h_j and k_j that reflect the relative influence of branch j . It is reasonable in most applications to suppose that one or more of the evaluation function values can be used to compose an influence measure, thereby yielding h_j and k_j values "automatically." This turned out to be the case for the nuclear refueling problem, as we show in the following.

Twelve evaluation functions were developed earlier for each k and t . Although we apply such functions only to those k and t that involve a violated discrete condition, this still yields too many evaluation functions to incorporate conveniently into target analysis for problems of the size of this application. We therefore conducted preliminary experiments with these evaluation functions on three small prototype problems to identify a good subset of these functions to use on larger problems. For these preliminary experiments, we chose the coefficients h_j and k_j for the discrimination analysis model by setting $k_j = \min(R_{s(j)}^2, E_{s(j)}^2)$ and setting $h_j = Mk_j$, where M is $1 + \max(k_j)/\min(k_j)$ times the larger of the two ratios $\#R_0/\#E_0$ and $\#E_0/\#R_0$. (By a theorem of Glover [5] this assures bounded optimality for the discrimination model.) Note the value for k_j is essentially that of measure (3) (divided by 2). The reason for this choice is the desire to have h_j and k_j reflect the influence of their associated branch, which by the previous discussion of influence is likely to be greater as the smaller of $R_{s(j)}$ and $E_{s(j)}$ increases (assuming the preferred branch is to drive the smaller value to 0). The use of the squared terms accentuates the relative degree of influence. Choosing $h_j = Mk_j$ attaches a penalty to incorrect classifications that is greater than the reward attached to correct classifications (where penalties and rewards affect the depth at which points lie outside or inside their associated half-spaces).

The result of the preliminary trials showed that the measures (1) and (5), the two ratios based on (3), and the function $(R_{kt} - E_{kt})/(R_{kt} + E_{kt})$ were significantly the most heavily weighted evaluation functions for discrimination analysis. Thus, the original twelve evaluation functions were reduced to five for the purpose of conducting target analysis on more moderate-sized problems.

As we undertook to apply target analysis for generating the vector x and scalar

b , we still, however, selected problems smaller than those we ultimately intended to solve, in order to keep calculations manageable. In this stage we again examined three test problems. We changed the values of the h_j and k_j coefficients slightly from our preliminary tests, however, by multiplying the original k_j values by the factor $1 - \min[|(R_{s(j)}/(R_{s(j)} + E_{s(j)})) - 0.3|, |(E_{s(j)}/(R_{s(j)} + E_{s(j)})) - 0.3|]$, and then determining the h_j values in relation to the k_j values as previously indicated. The effect of this multiplication is to give greater emphasis to those influential branches such that $E_{s(j)}$ and $R_{s(j)}$ are not equal, with a bias toward those where the indicated ratio is closer to 0.3 (in contrast to 0.5 when $R_{s(j)} = E_{s(j)}$). The reason for this is the greater uncertainty about a proper branching direction as $R_{s(j)}$ and $E_{s(j)}$ become more nearly equal—something to be avoided by an intelligent branching rule. (Thus the rationale accords with the rule described earlier for branching on values close to 0.3 or 0.7 in zero-one problems.)

3.4. COMPUTATIONAL RESULTS

Based on solving each of the three test problems to optimality, and then "re-solving" as previously indicated (selecting an optimal branch at each step), we produced the points A_j for discrimination analysis. During the re-resolution phase we selected those optimal branches with the largest k_j values, breaking ties arbitrarily. The discrimination LP model then produced the half-spaces for differentiating membership in the groups R_0 and E_0 . These learning and discrimination phases were exploited by using the resulting half-spaces to select branching variables and branching directions, in the application of branch-and-bound to the larger practical problems not previously examined. Finally, the algorithm employed network optimization techniques to exploit the underlying network structure of the problem.

Outcomes were extremely promising. Table 1 presents solution statistics for four versions of a test problem developed by the TVA. The sample utility system consists of five units: three 1000 MWe nuclear units, one 1000 MWe unit representing the non-nuclear balance of the system, and one 200 MWe unit

Table 1
Solution statistics on an IBM 4381

Number of time periods	Number of nodes	Number of arcs	Number of discrete requirements	Solution time	Number of sub-problems	Best sol. at sub-	Remaining levels problem No
10	44	110	30	1 sec.	20	5	0
20	84	220	60	7 min.	6056	2743	0
30	124	330	90	20 min.	8164	2852	6
42	172	462	126	30 min.	9129	257	7 *

* Solution value after 30 minutes was \$125,174,727; Solution value of MPSX after 7 hours was \$136,173,440.

representing interchange arrangements with other utilities. The planning horizon is partitioned into two-month-long time periods. The four versions of the problem correspond to four different planning horizons, having 10, 20, 30, and 42 time periods. In addition to the problem size and solution time, table 1 lists the number of subproblems solved in the branch-and-bound solution approach and the subproblem at which the best solution was found. It also lists the number of levels remaining unexplored in the branch-and-bound tree when the solution procedure stopped prior to verifying optimality due to imposed time limits. It is very encouraging that while the network relaxation is weaker than the standard LP relaxation (using the X_{kt} variables), the target analysis search procedure was able to find good solutions very early.

Solution times for the first three versions using Kazemersky's [17] mixed-integer programming model and MPSX on an 4381 were half an hour to two hours. The fourth version was run for seven hours using MPSX without reaching optimality. The best solution obtained had an objective function value of \$136,173,440. With a 30-minute time limit imposed on our discrete network solution procedure, a solution was obtained that was more than \$10,000,000 cheaper, with an objective function value of \$125,174,727.

5.0. Conclusions and directions for future research

The results of our study show that the use of a NETFORM model for the nuclear refueling problem, solved by means of a branch-and-bound procedure employing target analysis, was highly successful by comparison to previous efforts to solve this problem. While these results suggest the merit of our approach, it is appropriate to point out possible areas for refinement.

The structure of the overall procedure appears subject to improvement in several ways: 1) by including penalty calculations as parameters; 2) by identifying circumstances under which different, additional, hyperplanes should be generated (for example, generating different hyperplanes at different depths of the branch-and-bound tree or for different special subsets of variables); 3) by using sequentially generated hyperplanes in a conditional manner (for example, choosing one of the sets R_0 or E_0 at each step and generating a half-space that contains all points of this set that are not yet correctly classified, noting that all points in the opposite half-space will thus be classified correctly); 4) by a more thorough analysis of which branches should be selected in the "re-resolution" phase of target analyses; 5) by developing more refined rules for creating the h_j and k_j coefficients; and 6) by developing rules, as in Glover [3], allowing earlier branches to be re-adjudicated at later states of a branch-and-bound tree, and then deleted or reversed.

The ability of our approach to produce highly favorable outcomes lends encouraging support to this type of framework for blending the operations

research and artificial intelligence perspectives, and motivates additional research into the exploration of potential refinements.

References

- [1] L. Catalano, Innovative long-term scheduling plan aids nuclear plant retrofits (Iowa Electric Light & Power Co.), *Power* 128 (March 1984) 77-79.
- [2] P.F. Deaton, Utility system integration and optimization models for nuclear power management, Ph.D. Thesis, Massachusetts Institute of Technology, 1973.
- [3] F. Glover, Parametric branch and bound, *Omega* 6 (1978) 145-152.
- [4] F. Glover, Future paths for integer programming and links to artificial intelligence, *Comput. Opns. Res.*, 13 (1986) 533-549.
- [5] F. Glover, Improved linear and integer programming approaches for discrimination analysis, MSRS 88-1, University of Colorado, Boulder, Colorado 80308, 1988.
- [6] F. Glover, J. Hultz, and D. Klingman, Improved computer-based planning techniques, Part I, *Interfaces*, 8, No. 4 (1978) 16-25.
- [7] F. Glover, J. Hultz and D. Klingman, Improved computer-based planning techniques, Part II, *Interfaces* 9, No. 4 (1979) 12-20.
- [8] F. Glover, S. Keene and B. Dua, A new class of models for the discrimination problem, MSRS 86-5, University of Colorado, Boulder, Colorado 80308, 1986 Decision Sciences, to appear.
- [9] F. Glover and D. Klingman, Network application in industry and government, *AIIE Transactions* 9, No. 4 (1977) 363-376.
- [10] F. Glover and D. Klingman, The Simplex SON method for IP/embedded network problems, *Mathematical Programming Study* 15 (1981) 148-176.
- [11] F. Glover and C. McMillan, The general employee scheduling problem: an integration of MS and AI, *Comput. Opns. Res.* 13 (1986) 563-573.
- [12] G. Graves and R. McBride, The factorization approach to large-scale linear programming, *Mathematical Programming* 10, No. 1 (1976) 91-111.
- [13] C. Hamilton and W. Bingham, Management science applications in the planning and design of a water supply system for a nuclear power plant, *Interfaces* 9 (1979) pp. 50-62.
- [14] P.A. Jensen and J.W. Barnes, *Network Flow Programming* John Wiley and Sons, Inc., New York, 1980.
- [15] P.M. Kazemsky, A computer code for refueling and energy scheduling containing an evaluator of nuclear decisions for operation, Ph.D. Thesis, The Ohio State University, 1974.
- [16] P.M. Kazemsky, Scheduling optimal system operation using mixed integer programming, Proc. Nuclear Utilities Planning Methods Symposium, Chattanooga, Tennessee, January (ORNL-TM-4443).
- [17] P.M. Kazemsky, PODECKA - a pseudo one-dimensional point depletion algorithm, Proc. Nuclear Utilities Planning Methods Symposium, Chattanooga, Tennessee, January, 1974, (ORNL-TM-4443).
- [18] J. Kennington and R. Helgason, *Algorithms for Network Programming* (John Wiley and Sons, Inc., New York, 1980).
- [19] D. Klingman, S. Fuller and P. Randolph, (1976) A cotton ginning problem, *Operations Research* 24, No. 4 (1976) 700-717.
- [20] "Nuclear Services," Special Section of: *Electrical World* 200 (October 1986) N1-N18.
- [21] J.T. Rhodes, A model for the strategic operational planning of power systems containing large amounts of nuclear capacity, Ph.D. Thesis, Purdue University, 1972.
- [22] J.C. Turnage and B.E. Prince, A computer code (ORSIM) for the simulation and optimization of utility system operation with nuclear plants, Proc. Nuclear Utilities Planning Methods Symposium, Chattanooga, Tennessee, January 1974, (ORNL-TM-4443).

Copyright of Annals of Operations Research is the property of Springer Science & Business Media B.V.. The copyright in an individual article may be maintained by the author in certain cases. Content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.