INTERFACES
Vol. 9, No. 4, August 1979

0092-2102/79/0904/0012$01.25

IMPROVED COMPUTER-BASED PLANNING TECHNIQUES. PART II*†

Fred Glover

Professor of Management Science, College of Business,
University of Colorado, Boulder, Colorado 80302

John Hultz

Senior Systems Analyst, Analysis, Research, and Computation, Inc.,
Box 4067, Austin, Texas 78765

Darwin Klingman

Professor of Operations Research and Computer Sciences,
Center for Cybernetic Studies,
The University of Texas at Austin, BEB 608, Austin, Texas 78712

ABSTRACT. This is Part II of a two-part series. Part I showed how pure and generalized network models, and advances in methods of solving them, have resulted in dramatic cost savings for OR/MS practitioners. This paper focuses on network related formulation (NETFORM) models, which encompass an even wider variety of applications. We show how NETFORM has enabled the efficient solution of problems in scheduling, production, distribution, and other areas that were too large or difficult to be handled by previously applied techniques, including mixed integer programming.

Introduction

All too frequently Management Science practitioners discover that their major problems do not fit into straightforward linear programming frameworks, but instead involve nonlinearities expressed through discrete (integer programming) relationships. When this occurs, the inappropriateness of LP models typically causes a good deal of consternation. The history of discrete optimization abounds with situations where problems that would have been easy to solve as linear programs turned into computational monstrosities with the addition of discrete conditions.

---

*Part I appeared in *Interfaces*, Vol. 8, No. 4, pp. 16—25. This two-part paper is a condensed version of an original paper delivered at the SHARE XLVIII Conference in Houston, March 6-11, 1977. The full version of this paper may be obtained by writing Fred Glover or Darwin Klingman.

†This research was partly supported by Project NR047-172, ONR Contract N00014-78-C-0222 and Project NR047-C21, ONR Contracts N00014-75-C-0616 and N00014-75-C0569 with the Center for Cybernetic Studies, The University of Texas at Austin.

NETWORKS/GRAPHS — PLANNING

Recently, it has been discovered that a wide variety of discrete optimization problems have major network or network-related components. This invites the use of network models and solution techniques to replace the unsatisfactory approaches previously used. Moreover, advances in network-related formulation (or NETFORM) models have provided the ability to capture a still larger range of problems by means of representations that have an inherent network structure, making network solution technology applicable to these problems as well. The major purpose of this paper (Part II) is to present real-world applications that demonstrate the practical value of NETFORM models and the cost savings that result from their use.

This new NETFORM modeling technology, as noted in Part I of this paper, has several attractive features which lead to: (1) improved communication between practitioners and modelers because of the pictorial aspect of the models, (2) improved insight into the problem, making it possible to "see" where critical relationships lie and to interpret solutions obtained, and (3) an ability to solve many discrete optimization problems far more efficiently than in the past, including problems once believed to be too large or too complex to be solved within reasonable time limits.

The capacity of NETFORM models to render complex problems more solvable derives in large part from the significant advances in pure and generalized network solution methods, particularly in their efficient computer implementation. Part I of this paper discussed fundamental model concepts that led to real-world applications of the NETFORM modeling technique, and reported computational comparisons of network computer codes with a state-of-the-art commercial LP code.

Some NETFORM building blocks

The uses of arc multipliers described in Part I represent just a part of their full range of application. Introducing the requirement that flows on particular arcs must occur in integer (whole number) amounts makes it possible to model a much larger variety of applications, including problems such as assigning personnel to jobs where staffing requirements vary by time period, scheduling production on machines and assembly lines, scheduling payments on accounts subject to discrete payment levels, and determining optimal sizing and placement of electrical power stations.

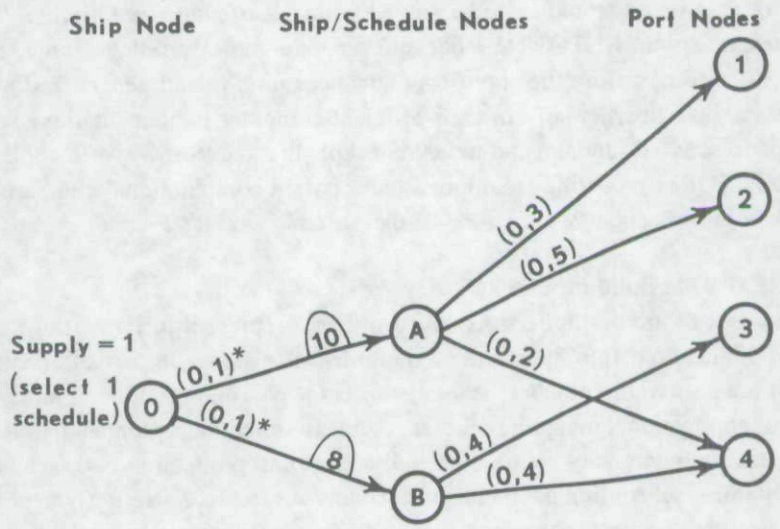
The power of NETFORM techniques, based upon the imposition of integer requirements in generalized networks, is illustrated by the fact that they enable the modeler to represent any 0-1 LP problem as an integer generalized network (GN) problem [3], [4]. Other NETFORM model components include arcs with "all-or-none" flows and side constraints. We will illustrate both the integer GN and the "all-or-none" model techniques in this paper. These techniques can also accommodate mixed integer 0-1 LP problems where the continuous part of the problem is a transportation, transshipment, or generalized network problem itself. An illustration in [7] shows how contemporary financial capital allocation problems can be modeled as integer GN problems. Other important real-world applications with this type of "mixed" structure include a variety of plant location models, energy models, physical distribution models, and dynamic production/distribution scheduling models.

The NETFORM representation is able to express rigorously all of the elements of these problems in a pictorial form. Consequently, it effectively *replaces* the

obscure and unilluminating algebraic representation by an equivalent, but much easier to understand, pictorial representation. But the advantages of the NETFORM approach do not end here. We have found that its underlying network-related structures can also be exploited by special solution methods that are substantially more efficient than the methods previously developed for the algebraic representations.

Figure 1 illustrates a useful modeling device commonly employed in the NETFORM approach. The relevant arc data are depicted by the same conventions employed in Part I. In particular, lower and upper bounds on the flow across an arc appear in parentheses, costs appear in squares, and multipliers appear in half ellipses. (Occasionally, we omit one or another of these notational components of an arc when it is not important to the discussion.) In addition, we introduce the convention that an *asterisk* above an arc indicates that its flow must be integer valued.

FIGURE 1. Ship Scheduling.



The setting for the example of Figure 1 is a ship scheduling problem. In general, such a problem would involve many ships, a variety of schedules for each, and numerous ports (each represented in several different time periods). Here we show the part of the model that applies to a single ship with exactly two schedules, A and B. Schedule A requires the ship to carry 10 tons of ore, which is distributed among the ports by dropping 3 tons at Port 1, 5 tons at Port 2, and 2 tons at Port 4. Schedule B requires the ship to carry 8 tons of ore, dropping 4 tons each at Ports 3 and 4. The diagram of Figure 1 provides a convenient way to visualize the specifications. It also provides a completely rigorous formulation of the problem — as rigorous as any algebraic formulation involving variables and constraints.

Specifically, the supply of 1 at the Ship Node indicates that the ship can select precisely one of its available schedules. (The upper bounds of 1 on the arcs leading to the Ship Schedule Nodes are, strictly speaking, superfluous, since neither of these two arcs can carry more than a unit flow due to the limited supply.) The asterisks,

which compel integer flows, rule out the possibility of sending a fractional flow value on either arc (i.e., selecting "part" of the associated schedule). The multipliers of 10 and 8 convert the arc flows into the number of tons required in schedules A and B, respectively. For example, if the flow on the arc to Ship Schedule Node A is 0, then $0 \cdot 10 = 0$, and therefore no flow gets transmitted to Node A for distribution to the ports (by means of this schedule). But if the flow on this arc is one, then the multiplier assures that 10 units of flow (tons of ore) are transmitted to A. Further, because the upper bounds on the arcs from A to the Port Nodes exactly sum to 10, each of these arcs must carry a flow equal to its upper bound, thereby assuring that the desired amounts are delivered to each of the ports. Arcs can be added emanating from the port nodes with lower and upper bounds limiting the total amount of ore received at each port during the time period from these and other ship schedules.

The same model structure can apply to many other situations, such as selecting among alternative schedules (or plans) for equipment purchase, real estate acquisition, portfolio investment, and so forth. A recent application of this type of model, which schedules Air Force pilots to advanced flight training courses [3], illustrates the power of specialized methods for such classes of NETFORM's. The standard mathematical programming formulation of this problem is a 0-1 integer programming (IP) problem with 460 0-1 variables and 520 constraints. An attempt by the Air Force to solve the problem with an IP solution routine was abandoned due to the prohibitive amount of computer time consumed in the solution effort. By contrast, a branch-and-bound approach specialized for the NETFORM structure (solving GN subproblems) normally obtains and verifies optimal solutions within 30 seconds on a CDC 6600.

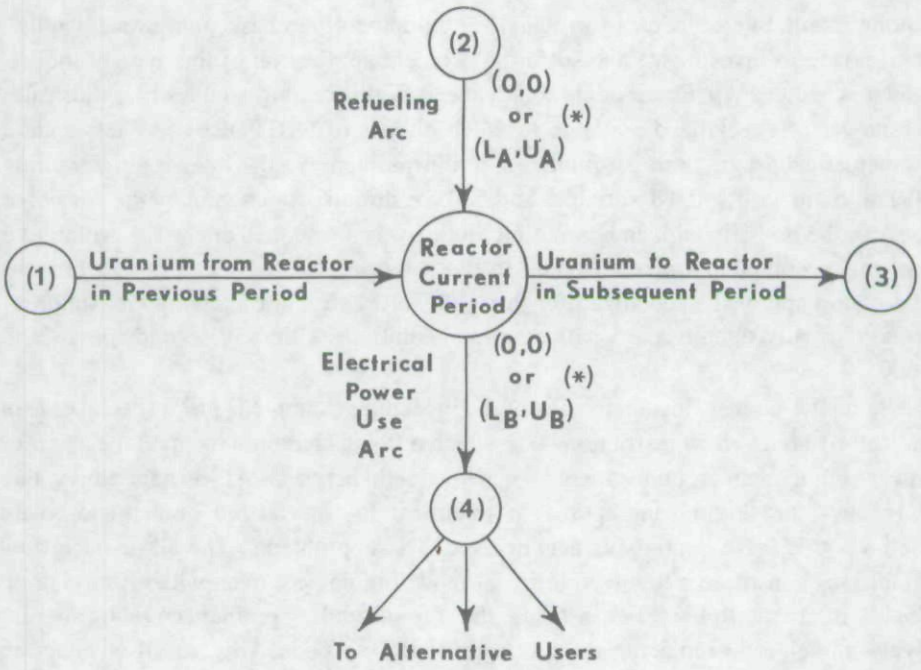
The 0-1 integer flow restriction of the preceding example is just a special case of an "all-or-none" flow restriction — in which a flow is required to equal the upper or lower bound on an arc, but cannot take any value in between. All-or-none flows, like 0-1 flows, are highly susceptible to treatment by specialized branch-and-bound methods that solve continuous network and GN subproblems. The all-or-none flow conditions constitute a highly significant modeling device, even without relying on the use of arc multipliers. For instance, the ship scheduling problem could be alternatively modeled by replacing the arc from node 0 to node A by an all-or-none arc having a lower bound of 0 and an upper bound of 10. Likewise, the arc from node 0 to node B would be replaced by an all-or-none arc with a lower bound of 0 and an upper bound of 8. The supply of node 0 would then be specified as bounded between 8 and 10. The following application provides another example of this technique.

Refueling nuclear reactors. The problem of determining the minimum cost refueling schedule for nuclear reactors is of considerable importance in the energy field. This problem was initially modeled by Kazmersky [5] as a mixed integer programming problem with no apparent connection to networks. However, after working closely with Dr. Kazmersky, we discovered a way to express the problem by a NETFORM representation that was not only equivalent to the original formulation, but also succeeded in reducing the size of each subproblem solved in the branch-and-bound procedure. The full transformation of the original problem to a NETFORM will not

be shown because the mathematics is somewhat intricate and the original formulation by itself consumes more than 20 pages of [5]. However, the key portion of the model, with alternative energy sources and additional network linkages omitted, is shown in Figure 2. In this diagram, a nuclear reactor in a given time period is depicted as a node with:

(1) an arc making available any unused uranium from the preceding period,
(2) an arc supplying new uranium as a result of refueling,
(3) an arc carrying unused uranium into the following period, and
(4) an arc carrying electrical power to the systems that require it in the current period.

FIGURE 2. One-Period Reactor Model.



(*)"Either-Or" Condition: Either the flow on the Refueling Arc is 0 and the flow on the Electrical Power Use Arc is at least L_B, or the flow on the Electrical Power Use Arc is 0 and the flow on the Refueling Arc is at least L_A.

This essential model component, repeated for each reactor and time period, is subject to the nonlinear (discrete) side condition that the reactor cannot be simultaneously refueled and used to provide electricity, i.e., the flow on either the Refueling Arc or on the Electrical Power Use Arc must be 0. Further, when the reactor is refueling, the flow on the Refueling Arc must be between a specified lower bound L_A and a specified upper bound U_A. Similarly, when the reactor is providing electricity to users, the flow on the Electrical Power Use Arc must be between a lower bound L_B and an upper bound U_B.

This NETFORM was used for solving the problem with the discrete "either-or" flow condition maintained external to the network. It was possible to do substantially better with this NETFORM than with the original mixed IP approach. A branch-and-bound procedure was developed, using the continuous relaxation of the NETFORM to generate network subproblems, and was applied to four versions of the nuclear refueling problem with data supplied by the TVA. The first three versions, while requiring half an hour to two hours to solve on an IBM 370/168 using MPSX, were easily solved in 5 to 20 minutes using the new code. The fourth version was by far the most difficult, involving 173 constraints, 126 0-1 variables, and 511 continuous variables. The original mixed integer formulation was run for seven hours on an IBM 370/168 using MPSX and then taken off the machine to avoid further computer run costs. At the end of this time, the best (minimum cost) solution obtained had an objective function value of $136,173,440. With a time limit of 30 minutes imposed on the solution effort, using the NETFORM, a solution was obtained that had an objective function value of $125,174,727, which *constitutes more than a $10,000,000 improvement.** Consequently, this application shows that the use of the NETFORM approach can provide improved solutions for problems too complex to be solved optimally by standard approaches within practical time limits.

Optimal lot sizing and machine loading for multiple products. The model to be described next is currently being used by a major manufacturing firm for large-scale task allocation. The problem objective is to minimize the combined costs of production and inventory holding by determining optimal product lot sizing and optimal assignment of production to machines.

The principal characteristics of the problem are:

1. The planning horizon is a single period, t weeks in length.
2. The products are designed to meet different needs and cannot be substituted for one another. Production of each product is a single-stage process.
3. Lot sizes are selected from a predetermined finite set of l possible lot sizes.
4. All lots of any single product must be produced on the same machine.
5. The machines work in parallel. They are similar in function, but they may differ in their rate and cost of operation. Some machines may be capable of producing several (or all) of the products while others may be more specialized.
6. The production capacities of all machines over the planning horizon are known constants. Each machine can produce only one product at a time.
7. Demand for each product is assumed to occur continuously at a known constant rate.

These characteristics give rise to the mathematical model shown in the Appendix.

The model is designed only to load the machines; it *does not schedule* the work on each machine. Rather, the main function of the model is to provide a capacity planning tool, allowing managers to retain the prerogative of determining the precise sequence and timing for implementing the candidate assignment over the horizon, in accordance with the objectives of this application. This provides flexibility to make

*Ed. Note: The reader is cautioned that the dollar values presented here are objective function values and *not* verified savings in accordance with *Interfaces* editorial policy.

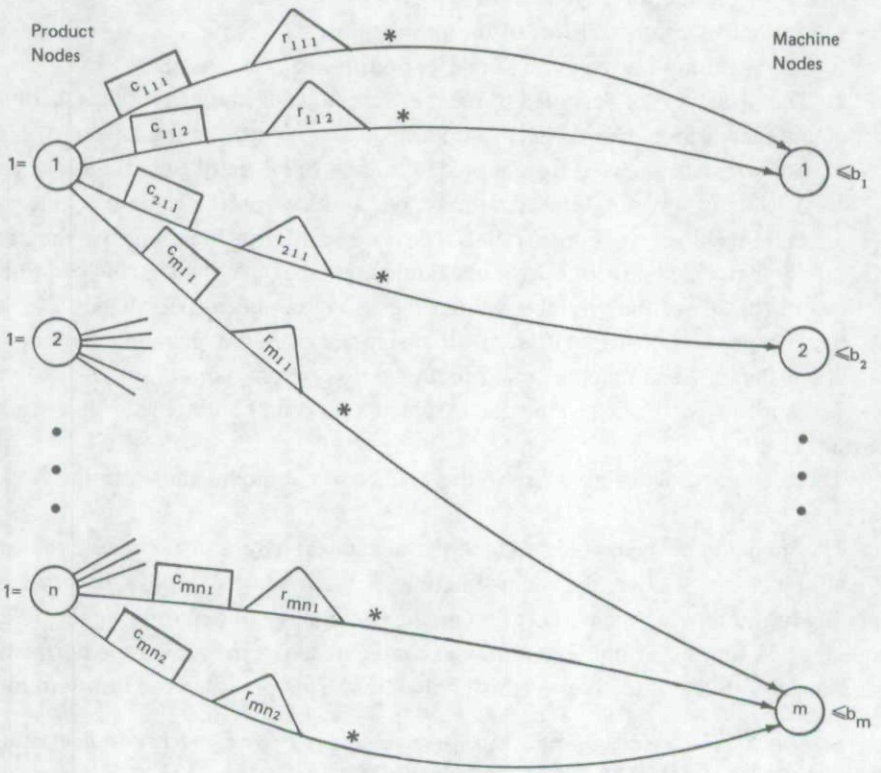
adjustments to special conditions and changed demands, while simultaneously aiding planning functions (such as evaluating the possible use of overtime shifts in periods when the candidate assignments tax weekly production capacities). For this type of flexibility and responsiveness to the needs of management, and to further support the analyses based on alternative assumptions of demands and capacities, it is especially important to be able to solve the model quickly for different (or recently updated) sets of data. Thus, the success of the application depends in large measure on the ability to solve the problem efficiently.

The firm in which this application arose initially tried to solve the problem using the 0-1 code (RIP30-C) developed by Geoffrion. This proved to be unsuccessful for two reasons: (1) the large array requirements of RIP30-C made it impossible to accommodate large problems, and (2) the method required excessive computation times even to solve problems with no more than 50 variables.

Consequently, it was apparent that an alternative solution approach was needed. The first step of our effort to identify such an approach was to characterize the network-related structure of the problem, which in this case turns out to be an instance of the 0-1 generalized network structure.

It was determined this problem can be represented graphically in the usual fashion by letting a node represent each equation and an arc each variable. Figure 3

FIGURE 3. Netform Representation.



illustrates the resulting graph. Two sets of nodes are created for this problem: a set of origin nodes, associated with products, and a set of destination nodes associated with machines. Since each product is to be scheduled on exactly one machine (and in exactly one lot size), the product nodes are each given a supply of 1. Consequently, an arc leading from a product to a machine will transmit a flow of 1 (the available supply) if the product at its tail node is to be produced on the machine at its head node in the lot size specified by the multiplier (in the triangle). Such a unit flow, therefore, transmits the lot size (via the multiplier) to the machine on which it is to be processed. If the flow is not equal to 1, it must be 0 as implied by the asterisk which limits flows to integer values. (Thus, an arc leading to machine i from product j for the kth possible lot size corresponds to the variable x_{ijk} defined in the Appendix.) The cost and multiplier of each of these arcs correspond to the cost and resource consumption of the associated variable. The upper limit on the demand at each machine node handles the restriction on the aggregate production capacity of the machine.

This graphical representation identifies the problem as a member of the class of problems known as *generalized assignment* problems. (If all the multipliers were 1, the problem would correspond to the classical network assignment problem.) This identification in turn leads to the selection of an appropriate solution method. In particular, extremely effective techniques for solving such problems have been developed by Ross and Soland [6] and imbedded in a computer code called BIG-A.

A comparison of the BIG-A code with the RIP30-C code for this problem shows that the BIG-A code is from 300 to 1000 times faster. In addition, the BIG-A code readily handles problems of up to 4000 variables within available computer memory. Thus, the firm now uses the graphical formulation coupled with the BIG-A code to solve the problem. This approach has made it possible to solve problems with 106 machines, 182 products, 4 lot-size options per machine/product combination, and 3772 0-1 variables, in .64 seconds on a CDC 6600 and 10 seconds on an IBM 370/145.

Conclusion

The preceding applications present a sampling of the kinds of modeling and solution capabilities that are emerging through the use of NETFORM techniques. Integer and mixed integer programming problems that have natural network components are prime candidates for the use of these powerful new tools. However, many other problems can also be tackled using these tools. It is worthwhile to note that the original formulations of the preceding applications bore no immediately apparent relationship to network problems. Nevertheless, the development of a NETFORM resulted in improved representational clarity and dramatically improved ability to solve the problems efficiently. On both of these grounds, the advances in NETFORM model techniques promise to usher in a new era of solution capability for integer and nonlinear problems that arise in practical settings.

Acknowledgement

We are especially grateful to Bud Curtin for comments and suggestions that have improved this paper.

References

[1] Charnes, A. and Cooper, W., *Management Models and Industrial Applications of Linear Programming*, Vols. I and II, Wiley, New York, 1961.

[2] Dantzig, G., *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.

[3] Glover, F. and Klingman, D., "Network Application in Industry and Government," *AIIE Transactions*, Vol. 9, No. 4, (1977), 363—376.

[4] Glover, F. and Mulvey, J., "Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks," MSRS 75-19, University of Colorado, Boulder, Colorado, 1975, to appear in *Operations Research*.

[5] Kazmersky, P., "A Computer Code for Refueling and Energy Scheduling Containing an Evaluator of Nuclear Decisions for Operation," Unpublished Dissertation, Ohio State University, 1974.

[6] Ross, T. and Soland, R., "A Branch-and-Bound Algorithm for the Generalized Assignment Problem," *Mathematical Programming*, 9 (1975), 91—103.

[7] Tavis, L., Crum, R., and Klingman, D., "Implementation of Large-Scale Financial Planning Models: Solution Efficient Transformations," Research Report CCS 267, Center for Cybernetic Studies, The University of Texas at Austin, 1976.

[8] Wagner, H., *Principles of Operations Research*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

Appendix. Mathematical model for the lot sizing and machine loading problem

A binary valued decision variable x_{ijk} is introduced which is defined to be 1 if product j is produced on machine i in the kth possible lot size and 0 otherwise. The combined set-up, production, and holding cost (per unit time) incurred when product j is produced on machine i in the kth possible lot size is denoted by c_{ijk}. Similarly, r_{ijk} denotes the capacity required on machine i to produce product j in the kth possible lot size. Finally, b_i denotes the aggregate production capacity of machine i over the t-week planning horizon.

The problem is to determine values for the variables that

$$\text{minimize:} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} c_{ijk} x_{ijk}$$

$$\text{subject to:} \quad \sum_{i=1}^{m} \sum_{k=1}^{l} x_{ijk} = 1 \text{ for } j = 1, \ldots, n, \tag{1}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{l} r_{ijk} x_{ijk} \leqslant b_i \text{ for } i = 1, \ldots, m, \tag{2}$$

$$x_{ijk} = 0 \text{ or } 1 \qquad \text{for } i = 1, \ldots, m; \quad j = 1, \ldots, m; \; k = 1, \ldots, l. \tag{3}$$

Constraints (1) and (3) together insure that one and only one machine and lot size combination is selected for each product. Constraint (2) insures that each machine is assigned production tasks commensurate with its capacity.