Real World Applications of Network Related Problems and Breakthroughs in Solving Them Efficiently

FRED GLOVER
University of Colorado
and
DARWIN KLINGMAN
University of Texas at Austin

Networks and network related problems occur with remarkable frequency in practical mathematical programming applications. This paper presents a variety of applications from industry and government that illustrate the scope and usefulness of network related formulations. In addition, recent breakthroughs in specialized methods and mathematical programming software systems that are capable of solving in only a few minutes problems that require many hours of computing time with commercial LP packages are reported. Finally, the latest developments in large scale applications are reported. These developments have made it possible to solve a manpower planning problem involving 450,000 variables in 26 minutes of central processing time on the IBM 360-65.

Key Words and Phrases: network applications, transshipment algorithms, machine independent software, computational testing, network problem formulations, fixed charge mathematical programming
CR Categories: 3.25, 3.57, 5.25, 5.32, 5.41

INTRODUCTION

A question of central concern to the mathematical programming practitioner is: What type of mathematical programming problems occur most frequently in the real world? It is admittedly an elusive sort of question, because it is often possible to "identify" a problem as being of one type, without realizing that the use of alternative formulation techniques or appropriate transformations will permit the problem to be represented in another form. (It often seems true that the easiest form of the problem to identify is the hardest one to solve!) Due to the significance of

A version of this paper was presented at Mathematical Software II, a conference held at Purdue University, West Lafayette, Indiana, May 29–31, 1974.
Authors' addresses: F. Glover, Department of Management Science, University of Colorado, Boulder, CO 80302; D. Klingman, Center for Cybernetic Studies, University of Texas at Austin, Austin, TX 78712.

the question, we have pursued an answer for a number of years, surveying mathematical programming applications reported in the literature, interviewing members of operations research teams from a variety of different corporations and government agencies, and swapping gossip with our friends and cohorts. As a result of these efforts, we have been led to the somewhat surprising conclusion that a very substantial proportion, perhaps as great as 70 percent, of the real world mathematical programming problems consist of, or can be transformed into, networks and network related problems. Specifically, the predominant number of practical mathematical programming applications appear to involve problems of the following types: assignment problems, transportation problems, transshipment problems, generalized transshipment problems, transshipment problems with extra linear constraints, integer problems whose relaxed problem is one of these, or a problem which is equivalent to one of these by a simple linear transformation.

As a result, in 1969 we began to concentrate our research on the development of new mathematical programming software that is computationally efficient for solving such problems, together with specialized techniques for transforming seemingly general linear of integer programming problems into one of these problems. The purpose of this paper is to summarize our accomplishments in this area. Since the development of this new software and the supporting techniques was motivated by a number of real world problems which we were asked to solve by corporations, government agencies, and nonprofit enterprises, another purpose of this paper is to summarize some of these real world problems.

APPLICATIONS

Applications arising in practical settings which we have found to be equivalent either to assignment, transportation, transshipment problems, or integer problems whose relaxed problem is one of these problems are briefly described below. Characteristically, in a number of these cases, the implicit network structures were not immediately visible. The techniques used to transform these applications into problems of the above forms are given in [1, 7, 8, 9, 10, 20, 22, 29, 30, 32, 37, 38].

1. A car distribution model whose goal is to determine how to assign car models to plant groups, i.e. to determine which models would be manufacturered in which plants and then to assign the cars from these groups to zones in the United States in order to meet customer demand at least cost, has been developed by General Motors. This is a transshipment problem whose size varies from 300 nodes and 1200 arcs to 1100 nodes and 3700 arcs. General Motors desired to solve such problems on a routine basis using a small computer with only 32K of central memory. A transshipment problem with 1100 nodes and 3700 arcs is a very large problem to solve in a reasonable time on such a computer since this is an LP with 1100 constraints and 3700 variables. Using our codes [13, 26] we were able to solve this problem in 37 seconds compared with 45 minutes using the GM transshipment code on the same computer.

2. A cash flow model is currently being developed by General Motors to handle their cash flow transactions. This model will identify the most economical way to manage receivables and payables; it is also relevant to the banking industry,

aiding in the planning of loans, purchases, and sales of securities, etc. This is a transportation problem with extra linear constraints; however, using the procedures in [9, 22, 30, 37] it is transformable into a transportation problem. The size of the problem is immense, involving 1000 origins, 1000 destinations, and 500,000 cells (arcs). Using our code [26] this problem can be solved in 30 minutes on an IBM 360/65. Allowing an optimistic estimate, a good commercial LP code such as the CDC OPHELIE code would require somewhat more than a 40-hour week of central processing time to solve this problem.

3. A model on "multiattribute" planning problems in personnel assignment is being used by the Office of Civilian Manpower Management of the U.S. Navy to assign personnel to jobs. The original formulation of this problem is a quadratic assignment problem. Using the procedure in [7] the problem can be transformed into an ordinary assignment problem.

4. A model is being used by Nabisco, Inc., for scheduling production and distribution of their cookies from production plants to regional warehouses, and from warehouses to local distribution facilities. The final formulation is a transportation problem.

5. A model is being used by the Texas Water Resource Management Development Board for scheduling dam reservoir levels in order to satisfy the peak seasonal demands of each region. This model is solved several thousand times each month in order to simulate all future contingencies over a 36-month horizon. There are two basic types of models for this problem. One is a transshipment problem and the other is a generalized transshipment problem. A generalized transshipment problem [3, 17, 23, 24, 34] is one where the flow on an arc may be increased or decreased as the flow traverses the arc (e.g. in this application, evaporation may cause the flow to decrease).

6. A manufacturing model is being used by Farah Manufacturing, Inc., which involves a two-stage transshipment model. The first stage involves the acquisition of cloth inventories from textile manufacturers. Once this inventory is ascertained it becomes the supply of the second phase which determines which pants are to be produced and which sew lines (production lines) are to cut these pants. The final formulation is a transportation problem having 150 origins, 1200 destinations, and 20,000 cells.

7. A model has been designed by the federal government to discretize a city into node areas and assign the people in these node areas to bomb shelters.

8. Models have been designed to assign people to parking spaces in a large city.

9. Models have been designed to solve the disposal waste problem, that is, to figure out where to locate dump areas, etc., in large cities. The formulation of this problem is a plant location whose relaxed subproblems are transshipment problems.

10. The Weyerhouser Lumber Company uses a distribution/production model for scheduling logs to plants, to finished products, to local market. The initial formulation is a chance-constrained linear programming problem whose deterministic equivalent can be transformed into a transportation problem using the procedure in [8, 9, 10].

11. A model is being used by the New Mexico Department of Agriculture for determining the time-phased planting, picking, storing, and distribution of cotton to gins. The "natural" formulation of this problem yields a mixed integer transportation problem with extra linear constraints. The transportation part of this formulation involves 2000 origins, 2300 destinations, and 2,460,000 cells. By exploiting the topological characteristics, the problem can also be formulated as a plant location transshipment problem consisting of 3,441 nodes, 61,640 arcs, and twenty 0-1 variables [29]. Coupling our code [26] with an efficient branch and bound procedure, we have been able to solve this problem in less than 1 hour of central processing time on a CDC-6600.

12. A rotation model is being used by the Navy Personnel Research and Development Center at San Diego for assigning enlisted personnel to sea and shore duty. This model gives rise to transportation problems that vary in size from 10 origins, 12 destinations, and 60 cells, to 1200 origins, 1800 destinations, and 750,000 cells. Using our code [26] we were able to solve a problem with 1093 origins, 1201 destinations, and 450,000 cells on an IBM 360/65 in 26 minutes of central processing time.

13. A model is being used by the Soviet Union for off-shore oil well drilling. The model selects locations for platforms and assigns wells to platforms. The model [2] is a plant location problem whose subproblems are transportation problems.

Other applications we have encountered include an ingredient distribution model used by Ralston-Purina, an aircraft allocation model, a student-professor-classroom assignment model, a tanker scheduling model, a telecommunication model used by the Texas State Department of Public Welfare, and an antifreeze distribution model used by Continental Oil Company.

COMPUTER CODE DEVELOPMENT AND COMPUTATIONAL TESTING

This section summarizes the historical background and principal findings of our code development and testing efforts. At some points the number of references to our work may seem disproportionate, but the fact is that no rigorously designed or extensive computational studies of network algorithms were undertaken prior to the investigations cited, and each of the developmental efforts back of these investigations led naturally, and in some cases inevitably, to the next.

We began jointly working with systems analysts in early 1970 in order to develop new techniques which are computationally efficient for solving assignment, transportation, transshipment, and generalized transshipment problems. Our first step in this direction was to elaborate on Johnson's "triple-label method" [14] by providing a method for characterizing successive basis trees with minimal relabeling. This procedure, which is called the augmented predecessor index (API) method, additionally indicates the most efficient way to coordinate the activities of finding the representation of the come-in arc (basis equivalent path), pricing out the basis, and updating basis labels. The API method has been a major contributor to the improvements in the computational efficiency of solution algorithms. Its use was a factor underlying the efficiency attained by the special purpose primal simplex transportation code of Glover et al. [15]. In addition, the procedure was incorporated into the Srinivasan-Thompson accelerated primal transportation code [36] and succeeded in cutting the solution times of that algorithm by more than half.

Using the API method, we undertook the development and computational testing of a primal transportation code from 1970 to 1972. This code was expressly designed for computational efficiency, but not at the expense of computer memory. The code was further designed for solving both capacitated and uncapacitated problems with nondense cost matrices (i.e. transportation problems where some cells may not be allowable). After spending 50 hours of central processing time on refining the algorithmic rules of the code, we found that the most efficient solution procedure arises by coupling the primal transportation algorithm with a version of the row minimum start rule and a modified row first negative evaluation rule. The Glover et al. study [15] revealed that this primal code was at least eight times faster than the Share out-of-kilter code [11, 35] and 150 times faster than OPHELIE/LP. Thus the o'd folklore about the superiority of out-of-kilter methods and a new folklore among computer service divisions about equivalence of general purpose and special purpose solution codes for transportation and transshipment problems were upended. Additionally, the study showed that this code requires much less memory than any other transportation code.

The largest problems solved in the study of [15] were 1000 origin by 1000 destination problems with an average solution time of 17 seconds. This study also tested the primal code on four computers, IBM 360/65, UNIVAC 1108, CDC 6400, and CDC 6600, in order to provide insights into conclusions based on comparing times on different machines and compilers. It was discovered that standard guidelines concerning the relative efficiencies of different computers were completely misleading, since the primal code ran only 10 to 12 percent faster on the CDC 6600 than on the UNIVAC 1108 and IBM 360/65, differing substantially from the estimates one would obtain by comparing instruction execution times specified for the machines.

Next we conducted extensive tests [27] on the effect of rectangularity (i.e. the proportionality of the number of origins and destinations), number of cells (variables), and objective function coefficient distribution, in order to provide users of the code and model builders with solution time estimators and other solution statistics related to structural characteristics of transportation problems. The studies showed that problem structure has an appreciable effect on the computational efficiency of the primal simplex transportation algorithm. The algorithm is sensitive not only to the total number of constraints, but also to the relative numbers of origin and destination constraints. In general, rectangular problems can be solved faster than square problems with the same number of constraints and variables. The row oriented start procedure and basis change criterion which yield the best results for square problems become even more efficient for rectangular problems. Additionally, the studies indicate that a change in the number of constraints affects total solution time to a greater extent than a change in the number of variables. Total solution time increases as the number of variables increases and as the variance in the cost coefficients increases.

Motivated by the fact that out-of-kilter codes were found to be substantially slower than the special primal code, Barr et al. [4] developed and coded an improved version of the out-of-kilter method. This code was compared against Clasen's SHARE code, Boeing's code, and the Texas Water Development Board code and found to be at least six times faster than the best of these (which differed from problem to problem). The study also examined a total of 215 capacitated and uncapacitated transshipment problems demonstrating the superiority of the improved version of the out-of-kilter code over the other out-of-kilter codes in all cases. The largest problems solved were 1500 node transshipment problems. The mean solution time was 34 seconds.

Still more recently, Glover et al. [13] have developed a general primal transshipment code. Computational comparison of this code with the out-of-kilter code by Barr et al. [4] reveals that the primal code is 30 percent faster on transshipment problems. This is rather startling since the code of [4] is probably the fastest existing out-of-kilter code and since conventional wisdom has it that labeling techniques are inherently more efficient than simplex techniques. The primal transshipment code was also tested against a nonsimplex code due to Bennington [5] and found to be eight times faster. This computational study also shows the superiority of the new primal transshipment code in terms of central memory requirements for storing network data. Specifically, the out-of-kilter codes require from $1\frac{1}{2}$ to 3 times the central memory requirements for storing network data as the primal transshipment code. The substantially increased problem size that can be accommodated in-core by the new primal code is illustrated in the study [13] by the solution of an 8000 node problem.

Concurrent with the development of the primal transshipment code [13], Glover et al. [17] developed a new list structure, called the augmented threaded index (ATI) method, for recording and updating the basis tree in adjacent extreme point network algorithms. The ATI method provides improvements in both network solution codes and computer sciences list structure which uses only two pointers per node while providing the user the ability to search a spanning tree both upward and downward. All previously proposed structures require at least three pointers per node. Additionally, computational testing of the ATI method shows that it improves the efficiency of our transshipment code [13] by at least 10 percent while requiring less computer memory.

All of this computational testing and research has culminated in the development of a large scale, in-core, out-of-core, special purpose, primal transshipment code by Karney and Klingman [26] which is capable of solving transshipment problems of almost unlimited size. This code has solved a problem for the Navy Personnel Research and Development Center in San Diego with 2400 nodes and 450,000 arcs on the IBM 360/65 and CDC 6600 using 26 minutes and 23 minutes of central processing time, respectively. Both machines used 40 minutes of peripheral processing time. Other classes of problems which we have developed computer codes to solve are generalized transportation and generalized transshipment problems [1, 6, 12, 17, 19, 20, 21, 23, 24, 34]. These generalized prob'ems may be viewed as similar to the classical (pure) transportation and transshipment problems except that flow on an arc (i, j) from node i to node j is subjected to amplification (or attenuation) by a factor p_{ij} (i.e. the amount flowing into node j along arc (i, j) is p_{ij} times the amount that leaves node i along arc (i, j)). A variety of practical economics problems, which in fact have nothing in common with "transportation," may be stated as such a problem (e.g. cash flow and budget models, electrical circuit models, plastic-limit analysis and design of structures problems, machine loading models, the classical cutting stock problem, blending models allocation of aircraft types to service routes, etc.). As shown in [1, 6, 12, 34] this seemingly slight generalization of the classical problems has a drastic effect upon the basis structure. However, by carefully analyzing and specializing the steps of the simplex method, we show in [17, 19] that the API method [14] (which we utilized in solving the pure problems) could be easily extended to handle the generalized problems. We have recently developed a code [23] based on this extension of the API method which, like the pure codes, is expressly designed for computational efficiency. Surprisingly, the generalized code requires very little computer memory beyond that required by the pure codes. Thirty hours of central processing time have been devoted to refining the algorithmic rules of the code, and it is able to solve generalized transshipment problems with 2000 nodes and 15,000 arcs in 50 to 60 seconds. This is approximately 100 times faster than state-of-the-art LP codes can solve these problems. Additionally, the generalized code is less than twice as slow on pure problems as our pure codes. Thus our generalized code is able to solve pure problems faster than the widely used codes for solving pure problems (except our own) and at the same time is able to solve more general problems.

CURRENT CODE DEVELOPMENT

Presently our computational research efforts are devoted to imbedding our network codes into more general algorithms for solving fixed charge transportation and transshipment problems, generalized transportation and transshipment problem with integer variables [33], transportation and transshipment problems with artibrary extra linear constraints [31], and multicommodity transshipment problems. Since the efficient solution of all of these problems rests primarily on efficient procedures for solving transportation, transshipment, generalized transportation, and generalized transshipment problems, we firmly believe that highly efficient codes can be developed for each of these problem classes. Preliminary results with small problems are encouraging. The outcome of more extensive testing will be reported in the near future.

REFERENCES

Note. References [16, 18, 25, 28] are not cited in the text.

1. Appa, G. M. The transportation problem and its variants. *Oper. Res. Quart. 24* (1973), 79–97.
2. Babayev, D. Mathematical model for optimal location of oil platforms and assignment of directed wells. Inst. of Cybernetics, Acad. of Sci. SSR, Baku.
3. Balas, E., and Ivanescu (Hammer), P. L. On the generalized transportation problem. *Manage. Sci. 11* (1964), 188–202.
4. Barr, R. S., Glover, F., and Klingman, D. An improved version of the out-of-kilter method and a comparative study of computer codes. To appear in *Math. Programming.*
5. Bennington, G. E. An efficient minimal cost flow algorithm. Oper. Res. Rep. 75, North Carolina State U., Raleigh, N. C. June 1972.
6. Charnes, A., and Cooper, W. W. *Management Models and Industrial Applications of Linear Programming, Vols. I and II.* Wiley, New York, 1961.
7. Charnes, A., Cooper, W. W., Klingman, D., and Niehaus, A. Static and dynamic biased

quadratic multi-attribute assignment models: solutions and equivalents. Rep. CS115, Cent. for Cybernetic Studies, U. of Texas at Austin, Tex.

8. CHARNES, A., GLOVER, F., AND KLINGMAN, D. A note on a distribution problem. *Oper. Res. 18*, 6(1970), 1213–1216.
9. CHARNES, A., GLOVER, F., AND KLINGMAN, D. The lower bounded and partial upper bounded distribution model. *Naval Res. Logist. Quart. 18* (1971), 277–278.
10. CHARNES, A., AND KLINGMAN, D. The distribution problem with upper and lower bounds on the node requirements. *Manage. Sci. 16*, 9 (1970), 638–642.
11. CLASEN, R. J. The numerical solution of network problems using the out-of-kilter algorithm. Memo. RM-5456-PR, Rand Corp., Santa Monica, Calif., March 1968.
12. DANTZIG, G. *Activity Analysis of Production and Allocation*, T. C. Koopmans, Ed., Wiley, New York, 1951, Ch. 23.
13. GLOVER, F., KARNEY, D., AND KLINGMAN, D. Implementation and computational study on start procedures and basis change criteria for a primal network code. To appear in *Networks*.
14. GLOVER, F., KARNEY, D., AND KLINGMAN, D. The augmented predecessor index method for locating stepping stone paths and assigning dual prices in distribution problems. *Transport. Sci. 6* (1972), 171–180.
15. GLOVER, F., KARNEY, D., KLINGMAN, D., AND NAPIER, A. A computational study on start procedures, basis change criteria, and solution algorithms for transportation problems. *Manage. Sci. 20*, 5 (1974), 793–814.
16. GLOVER, F., KARNEY, D., AND KLINGMAN, D. Double-pricing dual and feasible start algorithms for the capacitated transportation (distribution) problem. U. of Texas at Austin, Tex., 1970.
17. GLOVER, F., KLINGMAN, D., AND STUTZ, J. Extensions of the augmented predecessor index method to generalized network problems. *Transport. Sci. 7*, 4 (1973), 377–384.
18. GLOVER, F., AND KLINGMAN, D. Finding minimum spanning trees with a fixed number of links on a node. Presented at the 45th Nat. ORSA/TIMS Meeting, Boston, Mass., April 22–24, 1974.
19. GLOVER, F., AND KLINGMAN, D. A note on computational simplification in solving generalized transportation problems. *Transport. Sci. 7* (1973), 351–361.
20. GLOVER, F., AND KLINGMAN, D. On the equivalence of some generalized network problems to pure network problems. *Math. Programming 4*, 3 (1973), 369–378.
21. GLOVER, F., KLINGMAN, D., AND NAPIER, A. Basic dual feasible solutions for a class of generalized networks. *Oper. Res. 20*, 1 (1972), 126–137.
22. GLOVER, F., KLINGMAN, D., AND ROSS, G. T. Finding equivalent transportation formulation for constrained transportation problems. *Naval Res. Logist. Quart. 21*, 2 (1974).
23. GLOVER, F., KLINGMAN, D., AND STUTZ, J. Implementation and computational study of a generalized network code. Presented at the 44th Nat. Meeting of ORSA, San Diego, Calif., Nov. 12-14, 1973.
24. JEWELL, W. S. Optimal flow through network with gains. *Oper. Res. 10* (1962), 476–499.
25. JOHNSON, E. Networks and basic solutions. *Oper. Res. 14*, 4 (1966), 619–623.
26. KARNEY, D., AND KLINGMAN, D. Implementation and computational study on an in-core out-of-core primal network code. Res. Rep. CS 158, Cent. for Cybernetic Studies, U. of Texas at Austin, Tex.
27. KLINGMAN, D., NAPIER, A., AND ROSS, G. A computational study on the effects of problem dimensions on solution time for transportation problems. Res. Rep. CS 135, Cent. for Cybernetic Studies, U. of Texas at Austin, Tex., 1973.
28. KLINGMAN, D., NAPIER, A., AND STUTZ, J. NETGEN—a program for generating large scale (un)capacitated assignment, transportation, and minimum cost flow network problems. *Manage. Sci. 20*, 5 (1974), 814–822.
29. KLINGMAN, D., RANDOLPH, P., AND FULLER, S. A cottonpickin' cotton ginning problem. Presented at the 44th Nat. Meeting of the Operations Research Society of America, San Diego, Calif., Nov. 10–12, 1973.
30. KLINGMAN, D., AND ROSS, T. Finding equivalent network formulations for constrained network problems. Res. Rep. C.S. 108, Cent. for Cybernetic Studies, U. of Texas at Austin, Tex.

31. KLINGMAN, D., AND RUSSELL, R. On solving constrained transportation problems. To appear in *Oper. Res. Quart.*.
32. KLINGMAN, D., AND RUSSELL, R. The transportation problem with mixed constraints. To appear in *Oper. Res. Quart.*
33. KLINGMAN, D., AND STUTZ, J. Computational testing on an integer generalized network code. Presented at the 45th Nat. ORSA/TIMS Meeting, Boston, Mass., April 22–24, 1974.
34. LOURIE, J. Topology and computation of the generalized transportation problem. *Manage. Sci. 11* (1964), 177–187.
35. Out-of-kilter network routine. SHARE Distribution 3536, Share Distr. Agency, Hawthorne, N. Y., 1967.
36. SRINIVASAN, V., AND THOMPSON, G. L. Benefit-cost analysis of coding techniques for the primal transportation algorithm. *J. ACM 20*, 2 (April 1973), 194–213.
37. WAGNER, H. The lower bounded and partial upper bounded distribution model. To appear in *Naval Res. Logist. Quart.*
38. WAGNER, H. *Principles of Operations Research with Application to Managerial Decisions.* Prentice-Hall, Englewood Cliffs, N.J., 1969.

Received March 1974