
Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# Technical Note—Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks

Fred Glover, John M. Mulvey,

## ACKNOWLEDGMENTS

The author wishes to acknowledge the valuable comments of the referees, of Professor Heinz Isermann, and of his brother, Hans Roglin.

## REFERENCE

1. D. M. WARNER, "Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach," *Opns. Res.* **24,** 842–856 (1976).

# Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks

FRED GLOVER

*University of Colorado, Boulder, Colorado*

JOHN M. MULVEY

*Princeton University, Princeton, New Jersey*

(Received January 1977; accepted April 1979)

We show how to formulate a 0-1 integer programming problem as a "mixed integer" generalized network and as a discrete "0-$U$" pure network problem. Special integer programming structures allow convenient simplifications The usefulness of these formulations is in providing new relaxations for integer programming that can take advantage of recent advances in the development of efficient computer programs for network problems We cite three practical applications in which these ideas have led to marked improvement in solution efficiency.

THE MIXED INTEGER generalized network problem and a 0-$U$ (discrete) version of the pure network problem are capable of accommodating a variety of problems not ordinarily conceived of as related to networks, including the 0-1 integer programming problem. In this note, we will show more generally how to model the mixed problem

$$\text{Minimize } \textstyle\sum_{j\in N} c_j x_j \tag{1}$$

subject to

$$d_i \le \textstyle\sum_{j\in N} a_{ij} x_j \le b_i, \qquad i \in M = \{1, \cdots, m\} \tag{2}$$

$$u_j \ge x_j \ge 0 \quad \text{for} \quad j \in N \tag{3}$$

$$x_j \in \{0, 1\} \quad \text{for} \quad j \in N_1 \subset N \tag{4}$$

 

Published in Operations Research on June 01, 1980 as DOI: 10.1287/opre.28.3.829.
This article has not been copyedited or formatted. The final version may differ from this version.





where the set $M_j = \{i \in M\colon a_{ij} \neq 0\}$ has at most two elements for $j \in N - N_1$. (Thus (1), (2), (3) and (4) is an ordinary generalized network problem if $N_1$ is empty, and is a pure 0-1 integer programming problem if $N = N_1$.) We may also allow any subset of $x_j$ variables for $j \in N - N_1$ to be integer constrained. We first give a mixed integer generalized network formulation of this problem, and then give a discrete pure network formulation for the case in which the $N - N_1$ portion corresponds to a pure network.

## 1. THE GENERALIZED NETWORK FORMULATION

The mixed integer generalized network formulation of (1)-(4) may be obtained as follows. Note that if $M_j$ has at most two elements for each $j \in N_1$ (as well as for each $j \in N - N_1$), the problem is already a mixed integer generalized network. That is, each constraint of (2) may be associated with a node (having upper and lower bounds on its "demands" of $b_i$ and $d_i$), and each variable $x_j$ may be associated with an arc (whose endpoints are the nodes for which $a_{ij} \neq 0$).

For the general case, each constraint of (2) continues to correspond to a node, but a variable $x_j$ for $j \in N_1$ is viewed as an arc with multiple ends, one for each $i \in M_j$. To create the appropriate network structure, $x_j$ is subdivided into a collection of ordinary (generalized) arcs which link to each other through a common node $j_0$. Such a node $j_0$ is created for each $x_j$ and has no supply or demand of its own. In particular, for each $i \in M_j$ (when $|M_j| \geq 3$), an arc $(i, j_0)$ is created that connects the common node $j_0$ to "node $i$" (of the $i$th constraint). The conversion rules are as follows.

1. Assign arc $(i, j_0)$ a multiplier of $a_{ij}$ on its "node $i$" end.
2. Select some $i \in M_j$ to be designated by the symbol $j^*$, and designate arc $(j^*, j_0)$ to be a "0-1" arc. Give this arc a multipler on its $j_0$ end of $|M_j| - 1$, and a cost equal to $c_j$.
3. Assign each remaining arc $(i, j_0)$, $i \in M_j - \{j^*\}$, a multiplier of $-1$ on its $j_0$ end, a cost equal to 0, and a capacity (upper bound on its flow) equal to 1. (Each such arc will then automatically receive a flow value of 0 or 1 when arc $(j^*, j_0)$ receives such a value. Given unit capacities, this effect is also achieved by assigning $(j^*, j_0)$ any nonzero multiplier on its $j_0$ end, and assigning the remaining arcs multipliers of the opposite sign that sum to the negative of the multiplier for $(j^*, j_0)$.)

It is easy to see that the resulting mixed integer generalized network (with variables $x_j$ for $j \in N - N_1$ and for $j \in N_1$ with $|M_j| \leq 2$ corresponding to generalized arcs in the natural way) is equivalent to the original problem (1)-(4), since assigning an arc $(j^*, j_0)$ a flow of 0 or 1 accomplishes precisely the same effect as assigning the variable $x_j$ this value.

The usefulness of the mixed generalized network formulation is that it provides a new *relaxation* of problem (1)–(4) when the integer requirement is removed from the generalized network. This relaxation is less stringent than the ordinary linear programming relaxation of (1)–(4), but has the advantage of being much faster to solve (see e.g., [9]). Other characteristics of this relaxation will be discussed subsequently.

## 2. THE PURE NETWORK FORMULATON

The 0-$U$ pure network formulation of (1)–(4) can be described using the same terminology as the 0-1 generalized network formulation, under the assumption that the $N - N_1$ portion is a pure network. Here the problem (1)–(4) is first modified by the addition of a constraint

$$d_0 \le \sum_{j \in N_1} a_{0j} x_j \le b_0 \tag{5}$$

where the coefficients $a_{0j}, j \in N_1$, are selected so that

$$a_{0j} = -\sum_{i \in M_j} a_{ij}.$$

The constants $d_0$ and $b_0$ are selected so that (5) is redundant, e.g., $b_0$ can equal the sum of the positive $a_{0j}$ and $d_0$ can equal the sum of the negative $a_{0j}$. Thereupon, incorporating (5) into (2), the *amended* problem (1)–(4) has the property that $\sum_{i \in M_j^+} a_{ij} = -\sum_{i \in M_j^-} a_{ij}$ for all $j$ where $M_j^+ = \{i \in M_j\colon a_{ij} > 0\}$ and $M_j^- = \{i \in M_j\colon a_{ij} < 0\}$. For this problem the network is constructed as follows:

1. Create a node for each $i \in M$ as in the generalized network formulation.
2. Create two nodes, $j_1$ and $j_2$, for each variable $x_j$, $j \in N_1$, and an associated ordinary arc $(j_1, j_2)$ with capacity equal to $\sum_{i \in M_j^+} a_{ij}$. (Nodes $j_1$ and $j_2$ have no net supply or demand of their own.) Arc $(j_1, j_2)$ is designated a 0-$U$ arc, which means that it is restricted to receive either a 0 flow or a flow equal to its capacity. It is given a cost equal to $c_j$ divided by its capacity.
3. For each $i \in M_j^-$, create an ordinary arc $(i, j_1)$ with 0 cost and with capacity equal to $-a_{ij}$.
4. For each $i \in M_j^+$ create an ordinary arc $(j_2, i)$ with 0 cost and with capacity equal to $a_{ij}$.
5. If there is only a single arc $(i, j_1)$ entering node $j_1$, then this arc can be collapsed by designating node $j_1$ to be the same as node $i$. Similarly, if there is a single arc $(j_2, i)$ leaving node $j_2$, then this arc can be collapsed by designating node $j_2$ to be the same as node $i$.

The equivalence of this 0-$U$ pure network problem to the original 0-1 problem is established due to the fact that assigning an arc $(j_1, j_2)$ a flow equal to 0 or to its upper bound accomplishes the same effect as setting

$x_j$ equal to 0 or to 1, respectively. The relaxed problem in which the 0-$U$ restriction is removed, is a weaker relaxation than that of the generalized network formulation, but has the advantage that it can be solved still more efficiently (e.g., using the specialized codes of [2, 7, 12, 13]).

## 3. STRATEGIC CONSIDERATIONS AND SPECIAL STRUCTURES

In the generalized network formulation, it is legitimate to manipulate the costs on the arcs incident to a given node $j_0$ provided simply that these costs always sum to $c_j$. This is obvious by inspection, but it also may be interpreted as a form of "Lagrangian" manipulation [5, 6], taking side constraints into the objective function, where these side constraints stipulate that the flow on each arc incident to $j_0$ is to be the same. Moreover, by linear programming duality, there exists some such assignment of costs for which the optimum objective function value for the generalized network problem is the same as that for the direct linear programming relaxation of (1)–(4). Analogously, in the pure network relaxation, the costs on all arcs associated with a given variable $x_j$ can be manipulated so long as the weighted sum of these costs, each divided by the associated arc capacity, is equal to $c_j$. The Lagrangian interpretation again applies.

It should be noted that for special structures the form of these network relaxations can often be simplified. For example, in the case of the set covering and set partitioning problems (in which all $a_{ij} = 0$ or $1$ – see [1, 3, 11, 12]), the 0-$U$ formulation collapses all $j_1$ nodes into a single node, by Instruction 5. As a result, the relaxation can be modeled as an ordinary transportation problem simply by reversing the 0-$U$ slack arc relative to the upper bound $U$, thus giving the $j_2$ nodes a supply of $U$ and the common $j_1$ node a demand equal to the difference between the total supply and total demand. Srinivasan and Thompson [15] have proposed a network relaxation for set partitioning problems of exactly this form, and are to our knowledge the first to use ideas of this type.

In addition, special tricks exist to obtain both tighter and looser relaxations for certain multiple choice and scheduling structures. It is not our intent to itemize these tricks or the associated network strategies for capitalizing on their structures (e.g., by specialized penalty calculations), but merely to point out that a variety of alternatives exist for tailoring the general formulations to specific applications.

## 4. COMPUTATIONAL EXPERIENCE

From a computational standpoint, the utility of network-related models has been demonstrated in a variety of studies [4, 7, 8, 16]. We have had experience with three models, making use of the formulations proposed in this paper, that have proved to be highly susceptible to efficient solution in the network guise. The complete details of these models are rather extensive, due to their real world origins. Their parameters and structural characteristics are elaborated in [8–10, 12]. Having described the general model transformation in the preceding sections that led to the discrete network formulations of these problems, we will summarize here the computational results of employing these transformations.

The first of these models is a mixed integer programming problem to determine a minimum cost refueling scheduling for nuclear reactors [10]. Four versions of this problem were solved with data supplied by the TVA. The most difficult version of the problem (and the only one that was not solved easily in the network-related formulation) involved 173 constraints, 511 continuous variable, and 126 zero-one variables. The original mixed IP formulation was run for 7 hours with MPSX, at the end of which time the best (minimum cost) solution obtained had an objective function value of $136,173,440. With an imposed time limit of 30 minutes on the network-related solution effort, the best solution found has an objective function value of $125,174,727. The network-related approach used a simple LIFO branch and bound strategy to handle dichotomous conditions. The branching rule selected the discrete arc whose flow was closest to a 20% deviation from one of its bounds, and enforced a flow that satisfied the nearest bound.

The second type of problem examined in this framework was an Air Force flight training model involving the selection of schedules for assigning pilots to courses of study. The integer programming formulation for this problem involves 200 constraints on class sizes, 120 multiple choice constraints, and 460 zero-one variables. (See [9] for details.) The Air Force had attempted to solve this problem by customary integer programming approaches and concluded the problem was too difficult to solve optimally.

In view of this, a heuristic solution procedure had been developed for the problem in an attempt to get reasonably good solutions in an acceptable amount of time. Using the network-related model (again coupled with a branch and bound procedure), this problem was solved optimally in 10 seconds on a CDC 6600, which was substantially faster than the nonoptimal heuristic procedure (according to the Air Force and the developer of the heuristic procedure). The network related formulation of this problem involves 460 zero-one arcs with multipliers, 2,200 continuous flow arcs without multipliers, and 780 nodes. The 10-second solution time was again achieved by a simple LIFO branch and bound strategy. The branching rule in this case involved a fixed, *a priori*, ranking of the 0-1 scheduling arcs on the basis of their costs and of the costs of the second and third best alternatives. The time to set up this initial ranking is included in the 10-second solution time.

Finally, we have applied this type of network related model approach to a large (20,000 node, 100,000 arc) planning system for the United States Strategic Air Command. The integer programming formulation for this problem is a generalized set covering model with time constraints. With the network related model, the size of the problem effectively *decreases* because of a special structure for collections of columns in the mixed IP formulation. In particular, this structure admits $p$ unit entries in any of $q$ specified positions (where a single column may have repetitions of the coefficient organization over different subsets of $q$ positions, for different values of $p$ and $q$). These types of collections are readily modeled by the network related ideas previously described, allowing these collections to be implicitly defined (see [12]). In this manner, problems that cannot be handled by traditional optimization approaches are solved. Indeed, the usual zero-one IP model is enormous—100,000 rows, tens of millions of variables—and cannot be solved by linear programming. Yet, the discrete network model approach provides suboptimal solutions (within 5% of the optimal) using only the LIFO strategy coupled with a branching rule that first branches on arcs with smaller associated $M_j$ sets (see Section 2), and assigns a flow equal to the closest bound. The problem takes less than 30 minutes of execution time to solve.

The successes of the foregoing applications do not imply that the network related modeling ideas will be useful for all 0-1 problems. Srinivasan and Thompson found in their study [15] that network relaxations for set partitioning problems yielded computational successes only when the number of nonzeroes (in this case, 1's) did not exceed 3 or 4 per column, which suggests that the more general transformations may likewise be limited in value as the number of nonzeroes per column of the original formulation increases. Our experience with the preceding practical applications is compatible with this expectation, though the column densities averaged slightly higher than those found limiting for set partitioning.

The number of nonzeroes per column for the original form of the nuclear refueling problem comes closest to the 3–4 range: the columns for the integer variables each contain exactly four nonzeroes (although larger than 1), and the columns for the continuous variables contain from 2 to 4 nonzeroes, averaging 3. The Air Force flight training problem contains from 3 to 6 classes per schedule; hence, the original 0-1 integer formulation contains from 4 to 7 nonzeroes per column (all 1's), including the multiple choice constraints. The average number of nonzeroes per column for this problem is 5.3. The third application, i.e. the strategic planning system, possesses from 3 to 100 nonzero entries (1's) per column, averaging approximately 10 nonzeroes.

Quite recently, Nemhauser and Weber [14] have devised a relaxation for integer programming based on solving matching problems. Their relaxation is at least as strong as the LP relaxation. Thus, we are witnessing the emergence of network and graph (or matroid) theory relaxations that bracket the LP relaxations. Just as the network structure has been found to be highly exploitable, and yields attractive results for integer programs that exhibit a certain "structural proximity" to networks (in a manner as yet incompletely defined), so this emergence of relaxations based on additional structural characteristics may be expected to bring still other classes of integer programs into the range that can be solved effectively.

## ACKNOWLEDGMENT

We are indebted to David Karney and Darwin Klingman who have helped design and test the computer programs for the network related models of the nuclear refueling problem and the Air Force Flight Training problem whose details are elaborated in [8, 9]. Computation times reported for these problems are CPU seconds on the CDC 6600.

## REFERENCES

1. E. Balas and M. W. Padberg, "Set Partitioning," in *Combinational Programming: Methods and Applications*, B. Roy (ed.), D. Reidel Publishing Co., 1975.
2. R. S. Barr, F. Glover and D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes," *Math. Progr.* **7,** 60–86 (1974).
3. V. J. Bowman and J. H. Starr, "Set Covering by Ordinal Cuts; I. Linear Objective Functions," Management Sciences Report No. 321, Carnegie-Mellon University, 1973.
4. J. S. Dyer and J. M. Mulvey, "An Integrated Optimization/Information System for Academic Departmental Planning," *Mgmt. Sci.* **22,** 1332–1341 (1976).
5. M. L. Fisher and J. F. Shapiro, "Constructive Duality in Integer Programming," *SIAM J. Appl. Math.* **27,** 31–52 (1974).
6. A. M. Geoffrion, "Langrangean Relaxation for Integer Programming," *Math. Progr. Stud.* **2,** 82–114 (1974).
7. F. Glover, D. Karney and D. Klingman, "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," *Networks* **4,** 191–212 (1974).
8. F. Glover, J. Hultz and D. Klingman, "Improved Computer Based Planning Techniques—Part 2," *Interfaces* **9,** 12–20 (1979).
9. F. Glover, J. Hultz, D. Klingman and J. Stutz, "Generalized Networks: A Fundamental Computer-Based Planning Tool," *Mgmt. Sci.* **24,** 1209–1220 (1978).
10. P. Kazermersky, "Refueling and Energy Scheduling Containing an Evaluator of Nuclear Decisions for Operation," Ph.D. dissertation, Ohio State University, 1974.

11. R. MARSTEN, "An Algorithm for Large Set Partitioning Problems," *Mgmt. Sci.* **20,** 774–787 (1974).
12. J. M. MULVEY, "Special Structures in Network Models and Associated Applications," Ph.D. dissertation, University of California, Los Angeles, 1975.
13. J. M. MULVEY, "Testing of a Large-Scale Network Optimization Program," *Math. Progr.* **15,** 291–314 (1978).
14. G. L. NEMHAUSER AND G. M. WEBER, "Optimal Set Partitioning, Matchings, and Lagrangian Duality," Technical Report No. 399, SORIE, Cornell University, 1978.
15. V. SRINIVASAN AND G. L. THOMPSON, "Stopped Simplex Algorithm for the Set Partitioning Problem with Transportation-like Subproblems," Presented at the ORSA/TIMS Conference in Boston, April 24, 1974.
16. G. J. THOMPSON, "Recent Theoretical and Computational Results for Transportation and Related Problems," in *Modern Trends in Logistics Research*, pp. 388–415, W. H. Marlow (ed.), The MIT Press, Cambridge, Mass., 1976.

# A New Norm for Measuring Distance Which Yields Linear Location Problems

J. E. WARD
*Rensselaer Polytechnic Institute, Troy, New York*

R. E. WENDELL
*University of Pittsburgh, Pittsburgh, Pennsylvania*

(Received March 1977; accepted September 1979)

We propose a new norm, called the one-infinity norm, for characterizing distance in facility location problems. This new norm, which is a hybrid version of the rectilinear and Tchebycheff norms, not only gives a good characterization of distance (as compared, for example, to results by Love and Morris) but also has two alternate interpretations of travel. Furthermore, it yields linear programming formulations of location problems.

RECALL that a norm in $R^2$ is a function, $\|\cdot\|:R^2 \to R^1$, having the following properties: $\|\vec{v}\| = 0$ iff $\vec{v} = 0$; $\|\vec{v} + \vec{u}\| \leqq \|\vec{v}\| + \|\vec{u}\|$; $\|\alpha\vec{v}\| = |\alpha|\,\|\vec{v}\|$ for every scalar $\alpha$. In particular, for each $p \geqq 1$, an $l_p$ norm is defined as $\|\vec{v}\|_p \equiv (|v_1|^p + |v_2|^p)^{1/p}$, $\vec{v} = (v_1, v_2)$ so that we get the Euclidean norm when $p = 2$, the rectilinear (i.e., city-block or Manhattan) norm when $p = 1$, and the Tchebycheff norm when $p = \infty$, i.e., $\|\vec{v}\|_\infty = \text{Max}\{|v_1|, |v_2|\}$.

**Operations Research**
Vol. 28, No. 3, Part II, May–June 1980

0030-364X/80/2803-0836 $01.25