NETWORK RELAXATIONS AND LOWER BOUNDS FOR MULTIPLE CHOICE PROBLEMS

FRED GLOVER

University of Colorado, Boulder, Colorado

JOHN M. MULVEY

Princeton University, Princeton, NJ

ABSTRACT

This paper provides new network relaxations and penalty calculations for 0-1 integer programs with multiple choice constraints. The multiple choice problem, which may be viewed as a 0-1 problem with generalized upper bounding (GUB) restrictions, arises in a variety of applications involving scheduling, location and assignment.

The value of network relaxations lies in their speed of solution (roughly two orders of magnitude faster than LP relaxations) and in their numerical accuracy. Nevertheless, LP relaxations are more restrictive than network relaxations; that is, they provide tighter lower bounds. Hence the practical superiority of network relaxations appears to be chiefly for IP problems with no more than six or seven non-zeroes per column of the coefficient matrix. Consequently, it is important to identify new network relaxations with benefits for special structures and to derive stronger lower bounds for network relaxations generally, thus extending the range of problems network-based algorithms can solve advantageously.

The new relaxations of this paper are designed to have special benefits for multiple choice problems with non-negative or non-positive coefficient matrices. The new penalties provide stronger bounds for both the previous (more general) and the new (more specialized) relaxations. In addition, we derive penalties for the multiple choice sets that exhibit a special additive property not available to LP relaxations.

RÉSUMÉ

Cet article présente des nouvelles relaxations de réseaux et des calculs de pénalité pour programmes en nombres entiers 0-1 avec contraintes de choix multiple. Le problème à choix multiple, qui peut être vu comme un problème 0-1 avec restrictions de borne supérieure généralisée (GUB), apparait dans une variété d'applications comportant séquencement, situation et assignation.

La valeur des relaxations de réseaux repose dans leur vitesse de solution (en gros, plus rapides que les relaxations LP par deux ordres de grandeur) et dans leur précision numérique. Néammoins, les relaxations LP sont plus restrictives que les relaxations de réseaux; c'est-à-dire qu'elles fournissent des bornes inférieures plus serrées. Alors la supériorité pratique des relaxations de réseaux semble être limitée aux problêmes n'ayant pas plus que six ou sept entrées non nulles par colonne de la matrice des coefficients. En conséquence, il est important d'identifier de nouvelles relaxations de réseaux tirant bénéfice des structures spéciales et de dériver des bornes inférieures plus fortes pour les relaxations de réseaux en général, étendant donc le domaine des problèmes que les algorithmes basés sur des réseaux peuvent résoudre avantageusement.

Les nouvelles relaxations de cet article sont faites pour avoir des bénéfices spéciaux pour les problèmes à choix multiple avec matrices de coefficients non négatives ou non positives. Les nouvelles pénalités fournissent des bornes plus fortes à la fois pour les relaxations précédentes (plus générales) et nouvelles (plus spécialisées). En plus, nous dérivons des pénalités pour les ensembles à choix multiple qui exhibent une propriété additive spéciale qui n'est pas disponible pour les relaxations LP.

INFOR vol. 20, no. 4, November 1982

1 Problem Definition

This paper provides new network relaxations and branch and bound penalties for 0-1 integer programs with multiple choice constraints. The multiple choice problem arises in a variety of applications in scheduling, location and assignment.[4,6,9,15] Our penalty calculation results can also be applied to other types of IP problems, such as partitioning problems,[13] where it is possible to extract subsets of constraints of the multiple choice form.

We write the 0-1 integer program with multiple choice constraints as follows:

$$\text{Minimize } x_0 = \sum_{j \in N} c_j x_j \tag{1}$$

subject to:

$$d_i \leq \sum_{j \in N} a_{ij} x_j \leq b_i \quad i \in M = \{1, \dots, m\} \tag{2}$$

$$\sum_{j \in J_k} x_j = 1, k \in K = \{1, \dots, r\} \tag{3}$$

$$x_j \in \{0, 1\} \text{ for } j \in N = \{1, \dots, n\} \tag{4}$$

where the sets J_k, $k \in K$, form a partition of M. Any 0-1 integer program can be expressed in this form by adding slack variables to the inequalities $x_j \leq 1$ to create constraints of the form (3). However, we have segregated (3) to give special emphasis to the multiple choice structure it represents.

Problem (1), (2), (4) (and hence (1), (2), (3), (4)) has been modelled as a 0-1 generalized network problem and as a 0-U pure network problem (for varying values of U) in[11]. We will first summarize these formulations and then provide additional "pure network" formulations for non-negative coefficient matrices that take into account constraints (3). These new formulations have advantages over the pure network formulation of[11] when (3) is a non-trivial component of the 0-1 problem. Finally, we show that all of these formulations can be exploited in a unified penalty-calculation framework.

2 Formulations for General 0-1 Problems

The material of this section is primarily from[11]. It is included to provide a foundation for the new formulations of the next section, and to permit a statement of the main results of the final section that embraces both the more general and more special formulations.

Generalized network formulation

The 0-1 generalized network formulation of (1), (2), (4) may be described as follows. Note that (1), (2), (4) is already a 0-1 generalized network problem if there are two (or at most two) non-zero a_{ij} coefficients for each variable x_j. In this case each constraint of (2) corresponds to a node (with upper and lower bounds on its demand of b_i and d_i), and each variable x_j corresponds to an arc (whose endpoints are the two nodes for which $a_{ij} \neq 0$).

The more general case is handled as follows. Each constraint of (2) again corresponds to a node. Variable x_j is viewed as an arc with multiple ends, one for each constraint in which the variable has a non-zero coefficient. To accommodate the fact that an ordinary arc has only two ends, the variable x_j is "subdivided" into a collection of ordinary arcs which link to each other through a common node j_0 (which has no supply or demand of its own).

In particular, let $M_j = \{i \in M: a_{ij} \neq 0\}$. Then for each $i \in M_j$ (when $|M_j| > 2$) an arc (i, j_0) is created that connects the common node j_0 to "node i" (of the i^{th} constraint). The multiplier on the "node i" end is a_{ij}. The multiplier on the "j_0" end is -1 for all but one of the arcs, which will be called arc (i^*, j_0), where i^* is one of the indices $i \in M_j$. Arc (i^*, j_0), which may be selected arbitrarily, is given a positive multiplier on its j_0 end equal to $|M_j| - 1$. Finally, (i^*, j_0) is designated to be a "0-1" arc with a cost of c_j, while the other arcs are given upper bounds of 1 and costs of 0. Each of these other arcs will automatically receive a value (flow) equal to 0 or 1 when the (i^*, j_0) arc receives that value. (The effect can more generally be achieved by assigning (i^*, j_0) any non-zero multiplier, and allowing the other arcs to have any multipliers of the opposite sign that sum to the negative of the multiplier for (i^*, j_0).)

The pure network formulation

The 0-U pure network formulation of (1), (2), (4) can be described using the same terminology as the 0-1 generalized network formulation. The designation 0-U comes from restricting the flow on certain arcs to either their lower bound (0) or their upper bound (U). Here the problem (1), (2), (4) is first modified by the addition of a constraint

$$d_0 \leq \sum_{j \in N} a_{0j} x_j \leq b_0, \tag{5}$$

where the coefficients a_{0j}, $j \in N$, 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, for example, 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), (2), (4) has the property that

$$\sum_{i \in M_j^+} a_{ij} = - \sum_{i \in M_j^-} a_{ij}$$

where $M_j^+ = \{i \in M_j: a_{ij} > 0\}$ and $M_j^- = \{i \in M_j: a_{ij} < 0\}$. 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 $x_j, j \in N$, and an associated ordinary arc (j_1, j_2), with capacity

$$u_j = \sum_{i \in M_j^+} a_{ij}.$$

   (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 u_j. 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, 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, 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 u_j accomplishes the same effect as setting x_j equal to 0 or to 1, respectively. The relaxed problem, in which 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[2,8,14].

Lagrangean manipulations

In the generalized network formulation the costs on the arcs incident to a given node j_0 can be manipulated provided that these costs always sum to c_j. This may be interpreted as a form of "Lagrangean" manipulation,[5,7] where the side constraints stipulating that the flow on each arc incident to j_0 is to be the same are taken into the objective function. It can be shown that there exists some such assignment of costs for which the optimum objective function value for the generalized network problem is identical to that for the usual linear programming relaxation of (1), (2), (4). Likewise, in the pure network formulation, the costs on 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.

Allowing a heavier reliance on Lagrangean manipulations, a somewhat simpler type of constrained network formulation is immediately available that applies to non 0-1 as well as to 0-1 problems. This formulation arises by splitting the non-zeroes of any column into disjoint pairs (with perhaps one element left over). Each pair corresponds to an arc of a generalized network (singletons corresponding to slack arcs). The original problem is recovered by requiring the flows on each of these arcs to be the same.

Since the equal-flow side constraints destroy the network structure, the imposition of the integer restrictions does not provide a network equivalent. The network portion of the problem constitutes a valid relaxation, however, and a Lagrangean strategy can be used to capture some of the influence of information contained in the side constraints.

This simple splitting of the columns into paired elements (handling the side constraints by Lagrangean relaxation) introduces no new continuous variables, but $[K/2]$ new *integer* variables, where K is the number of non-zero coefficients in the original non-network portion of the problem. (By contrast, the preceding 0-1 generalized network formulation requires roughly K new *continuous* variables (arcs), but no new integer variables.)

Nemhauser and Weber[(15)] apply an instance of such a column splitting scheme to the classical partitioning problem – with a novel twist. In this approach, the integer restrictions are included within the generalized network relaxation itself. Redundant upper bound constraints are introduced where necessary, so that each column splits perfectly into a collection of paired non-zeroes (both equal 1), thereby producing a weighted matching problem. The side constraints must of course still be handled by Lagrangean manipulation.

In settings more general than the matching problem, the need to introduce new integer variables and the dependence upon the side constraints to achieve equivalence when the integer conditions are enforced make the column splitting formulation appear less attractive than the previous formulations. In addition, there seems to be no advantageous way to specialize the column splitting formulation to the multiple-choice problem. Nevertheless, the penalty calculation results of section 4 are applicable to this formulation as well as to the others.

3 New Formulations for the Multiple Choice Problems

We now provide two alternative pure network formulations for the 0-1 multiple choice problem. Each is designed to take explicit account of the multiple choice constraints for the case in which $M_j = M_j^+$ for all $j \in J_k$, or $M_j = M_j^-$ for all $j \in J_k$ for all $k \in K$. We suppose that the pure network formulation of[(11)], just described, is first applied to the partial problem (1), (2), (4) in which the constraints of (3) are *not* incorporated into those of (2). It suffices to assume $M_j = M_j^+$ for all $j \in J_k$, since an inequality of (2) with non-positive coefficients can be multiplied through by -1 to produce an inequality with non-negative coefficients. For this case the node j_i is made a common node for all $j \in J_k$. We now show how to add constraints (3) to this formulation.

Alternative 1: Let w_k be the maximum capacity of the arcs (j_1, j_2) for $j \in J_k$. Create a dummy node i_0 with unrestricted supply and demand (corresponding to the creation of a new redundant inequality for (2)). For each $j \in J_k$ such that arc (j_1, j_2) has capacity less than w_k, add a zero cost arc (j_2, j_0) with capacity equal to w_k minus the capacity of arc (j_1, j_2). Finally, set the capacity of (j_1, j_2) equal to w_k for all $j \in J_k$, and give the common node j_1 a supply of exactly w_k.

Alternative 2: Identify w_k and create a node i_0 as in alternative 1. Also, let l_k be the minimum capacity of the arcs (j_1, j_2) for $j \in J_k$. Give the common node j_1 a supply of exactly w_k. If $l_k \neq w_k$, create a zero-cost arc (j_1, i_0) with capacity equal to $w_k - l_k$. Finally, impose the stipulation that exactly one of the arcs (j_1, j_2) for $j \in J_k$ must have a flow equal to its capacity, and all other (j_1, j_2) arcs must have a 0 flow.

It should be noted that the stipulation of alternative 2 is automatically accommodated in alternative 1 by the 0-U requirement on the arcs (j_1, j_2) for $j \in J_k$. For this reason it is plausible to suppose that the relaxation provided by alternative 1 (dropping the 0-U requirement) would be stronger than the relaxation provided by alternative 2 (dropping the 0-U requirement and the added stipulation). However, neither relaxation dominates the other in all cases.

Because they are explicitly designed to accommodate the constraints of (3), these relaxations are often strongly preferable to those of[11] in the multiple choice context. Still, it should be borne in mind that the relaxation provided by the 0-1 generalized network formulation can be organized so that it is more restrictive than either of these alternatives. This occurs by designating the node j^* to be the node $i \in M_j$ that corresponds to the multiple choice constraint containing the variable x_j. For this reason, when the conditon $M_j = M_j^+$ does not hold for all $j \in J_k$, the 0-1 generalized network formulation is likely to strongly dominate the 0-U pure network formulation for multiple choice problems. (It is possible to construct variants of alternatives 1 and 2 to handle the case for $M_j^+ \neq M_j$ by imposing a further stipulation that flows on pairs of arcs must be equal, but the relaxations provided by these variants seem particularly weak).

4 Network Penalty Calculations

Each of the preceding formulations can be exploited in a branch and bound setting by specifying up and down lower bounds ("penalties"), specialized to the multiple choice problem, for compelling a 0-1 variable to take the value of 1 or 0, respectively. The multiple choice sets in each case receive imputed penalties that are additive – that is, penalties whose sum gives a valid bound on the difference between the optimum x_0 value for the network relaxation and the optimum x_0 value for the original multiple choice problem.

Definition: Relative to a given variable x_j of the multiple choice problem, and a given problem formulation (to be specified), E_j is the set whose elements are as follows.

- For the generalized network formulation: the arcs incident to node j.
- For the pure network formulation: the arcs incident to nodes j_1 and j_2 when M_j^+ and M_j^- are non-empty; the arcs incident to j_1 when M_j^+ is empty; and the arcs incident to j_2 when M_j^- is empty.
- For the column-splitting formulation: the arcs derived from the paired non-zeroes of column j.
- For the multiple choice network formultions of alternatives 1 and 2: the arcs incident to node j_2.

Remark: Every arc of E_j receives a 0 flow when $x_j = 0$ and receives a flow equal to its upper bound when $x_j = 1$. (The result is immediate by analysis of the constructions of the preceding sections.)

The identification of E_j provides a basis for specifying penalties applicable to all of the preceding formulations by means of a single, unifying collection of formulas. For this purpose we require some additional notation. Denoting an arbitrary arc of the network by the symbol e, and let U_e = the upper bound (capacity) of arc e. Then, relative to the optimal linear programming solution to a relaxed network problem (dropping 0-1 and 0-U requirements, plus any related side conditions), let:

UB = $\{e$: e is non-basic at its upper bound $U_e\}$
LB = $\{e$: e is non-basic at its lower bound $0\}$
R_e = the absolute value of the reduced cost (updated linear programming objective function coefficient) of arc e.

Theorem 1

For a given set J_k, and a variable x_h, $h \in J_k$, valid penalties $P_h(1)$ and $P_h(0)$ for the assignments $x_h = 1$ and $x_h = 0$, are given by:

$$P_h(1) = \sum_{e \in S_h} U_e R_e,$$

where

$$S_h = \{e: e \in \text{LB} \cap E_h \text{ or } e \in \text{UB} \cap E_j \text{ for some } j \in J_k - \{h\}\}$$

and

$$P_h(0) = \min_{j \in J_k - \{h\}} (P_j(1)).$$

Proof: The assignment $x_h = 1$ yields upper bound flows U_e on all arcs $e \in E_h$. The LP reduced cost is 0 for all arcs with flows strictly between their bounds, and flows already at upper bounds do not incur additional cost. Consequently, the cost increase due to changed flows over arcs e in E_h equals (or exceeds) $U_e R_e$ summed over arcs $e \in$ LB $\cap E_h$. In addition, setting $x_h = 1$ compels $x_j = 0$ for all $j \in J_k - \{h\}$. This in turn yields 0 flows for all arcs e in the sets E_j, for $j \in J_k - \{h\}$. Combining these cost increases yields the formula for $P_h(1)$. On the other hand, setting $x_h = 0$ implies $x_j = 1$ for some $j \in J_k - \{h\}$. Consequently, the cost increase due to this assignment is bounded below by the value of $P_h(0)$. This completes the proof.

Theorem 1 identifies the form of the penalty calculation available to the network relaxation by restricting attention to a 0 or 1 assignment for a single variable. Next, we show how to obtain imputed penalties for the multiple choice sets that exhibit the additive property referred to earlier.

Theorem 2

Let the penalty P_k for the multiple choice set J_k be give by

$$P_k = \operatorname*{Min}_{j \in J_k} (P_j(1))$$

and let $x_0{}^*$ and $x_0{}'$ respectively denote the optimum x_0 values for the multiple choice problem and its network relaxation.

Then

$$\sum_{k \in K} P_k \leq x_0{}^* - x_0{}'.$$

Proof: Some x_j for $j \in J_k$ must be assigned the value 1. By the validity of the $P_j(1)$ penalties (established by theorem 1), it follows that $P_k \leq x_0{}^* - x_0{}'$ for each $k \in K$. Furthermore, the $P_j(1)$ penalties do not require an implicit basis exchange for their calculation, but are all derived relative to the current updated network (LP) representation of x_0. The additive property of the P_k penalties follows at once.

The preceding theorem has two noteworthy aspects. First, from the identification of $P_j(0)$ in theorem 1, it follows that

$$P_j(0) = P_k \quad \text{for all } j \in J_k - \{k^*\},$$

where k^* is an index in J_k for which $P_{k^*}(1) = P_k$. From the standpoint of branch and bound choice rules, the assignment $x_{k^*} = 1$ represents the best (least penalty) branch available from the set J_k. A standard branching strategy therefore would elect to set $x_{q^*} = 1$, by the imposition of the corresponding flows in the relaxed network problem, where the index q is identified by

$$P_q = \operatorname*{Max}_{k \in K} (P_k).$$

Second, the LP analog of the value P_q provides the limiting bound on the objective function increment in customary LP relaxations of (1)–(4). More precisely, letting p_k* denote the multiple choice penalty value associated with set J_k in the LP relaxations, the standard LP bound condition has the form

$$\max_{k \in K} (P_k^*) \leq x_0^* - x_0',$$

as contrasted to the additive bound condition expressed in theorem 2.

References

(1) E. Balas and M.W. Padberg, "Set partitioning," in B. Roy, ed., *Combinatorial Programming: Methods and Applications.* 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," Mathematical Programming, vol. 7, no. 1, 1974, 60–86.

(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. Dyer and J. Mulvey, "An integrated optimization/information system for academic departmental planning," Management Science, vol. 22, no. 12, August 1976.

(5) M.L. Fisher, W.D. Northup, and J.F. Shapiro, "Using duality to solve discrete optimization problems: theory and computational experience," Mathematical Programming Study, vol. 3, 1975, 56–94.

(6) R.S. Garfinkel and G.L. Nemhauser, *Integer Programming.* New York: John Wiley & Sons, 1972.

(7) A.M. Geoffrion, "Lagrangean relaxation for integer programming," Mathematical Programming Studies, vol. 2, 1974, 82–114.

(8) 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, vol. 4, 1974, 191–212.

(9) F. Glover and D. Klingman, "Network application in industry and government," AIIE Transactions, December 1977, 363–76.

(10) F. Glover, J. Hultz, D. Klingman, and J. Stutz, "Generalized networks: a fundamental computer-based planning tool," Management Science, vol. 24, no. 12, August 1973, 1209–20.

(11) F. Glover and J.M. Mulvey, "Equivalence of the 0-1 integer programming problem to discrete generalized and discrete pure network models," Operations Research, vol. 28, no. 3, May-June 1980, 829–36.

(12) Philip Kazermersky, "Refueling and energy scheduling containing an evaluator of nuclear decisions for operation," PH D thesis, Ohio State University, 1974.

(13) R. Marsten, "An algorithm for large set partitioning problems," Management Science, vol. 20, no. 5, 1974, 774–87.

(14) J.M. Mulvey, "Testing of a large-scale network optimization program," Mathematical Programming, vol. 15, 1978, 291–314.

(15) G.L. Nemhauser and G. Weber, "Optimal set partitioning, matchings, and Lagrangean duality," Technical Report No. 399, College of Engineering, Cornell University, Ithaca, NY, Dec. 1978.

(16) H.M. Salkin, *Integer Programming.* Addison-Wesley Publishing Co., 1975.

(17) 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, Mass., 24 April 1974.