SOLUTION OF CONSTRAINED GENERALIZED TRANSPORTATION PROBLEMS USING THE PIVOT AND PROBE ALGORITHM

by

A.P. Sethi & G.L. Thompson

December, 19R3

DRC-7O-17-B?

SOLUTION OF
CONSTRAINED GENERALIZED TRANSPORTATION PROBLEMS
USING THE PIVOT AND PROBE ALGORITHM

by

Awanti P. Sethi*
and
Gerald L. Thompson*[1]"

December 1982

•**Economics & Management Dept, Rhode Island College, Providence, RI 02908**

[+]**Graduate School of Industrial Admn., Carnegie-Mellon University, Pittsburgh, PA 15213**

This report was prepared as part of the activities of the Management Science Research Group, Carnegie-Mellon University, under Contract No. N0Q014-75-C-0621 NR 047-048 with the U.S. Office of Naval Research. Reproduction in whole or part is permitted for any purpose of the U.S. Government

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pa 15213

Solution of Constrained Generalized Transportation Problems Using The Pivot-and-Probe Algorithm

By

Awanti P. Sethi and Gerald L. Thompson

Abstract

In this paper we use a specialized version of our pivot and probe algorithm to solve generalized transportation problems with side constraints. The dual of an m x n generalized transportation problem with t side constraints is a linear program with m + n + t variables and up to m x n constraints. We solve the dual problem using the probe operation to select only the most important constraints to consider. We present computational experience on problems of sizes up to 180 x 180, having various degrees of density and having as many as 10 side constraints. It was found that for a given size and density, problems become harder to solve as the number of side constraints increases. Also, for a fixed number of side constraints, the solution difficulty increases with size and density. We found that our method was able to solve problems of the quoted sizes relatively quickly, with relatively few pivots, and without using basis reinversion.

Keywords:

Generalized Transportation Problem
Linear Programming
Simplex Method
Candidate Constraints
Side Constraints
Probe Operation

1 *INTRODUCTION*

The Generalized Transportation Problem (GTP) was introduced by Fergusan and Dantzig [13] in their study of an aircraft routing problem. Eisemann and Lourie [12] applied it to the machine loading problem which we discuss briefly in Section 2. The loop-technique of the stepping-stone algorithm for the ordinary transportation problem was extended to the GTP in [5] by Balas and Ivanescu and other theoretical results were also given there. Eisemann [11] and Lourie [24] also gave further results concerning the topology of a feasible solution to a GTP. Balas [6] gave some post optimization results and methods for including additional constraints. In Volume 2 of [9] Charnes and Cooper treated the GTP from the point of view of dyadic models and sub-dual methods. Balachandran and Thompson [1, 2, 3, 4] derived the operator theory of parametric programming for the GTP.

In the 1970's a number of authors presented efficient algorithms for solving network problem, see Bradley, Brown and Graves [7], Glover, Karney and Klingman [16], and Srinivasan and Thompson [28, 29]. Other authors extended the work to network problems with side constraints, see Chen and Saigal [10], Glover, Karney, Klingman and Russell [17], Glover, Hultz and Klingman, [14], Klingman and Russell [23]. For work on generalized network problems with side constraints, see Glover and Klingman [18], Hultz and Klingman [22], Glover, Hultz, Klingman, and Stutz [15], Helgason and Kennington [21], and McBride [25], Brown and McBride [8], and Gupta and McBride [20].

In this paper we use the pivot-and-probe algorithm (PAPA) which we introduced in [27] to solve uncapacitated GTP having some side constraints. We take advantage of the fact that the dual of an uncapacitated GTP has fewer variables than constraints, and use the probe operation to select only the most important constraints to consider. We present computational experience which indicates that the PAPA algorithm works better on sparse than dense problems, and better with fewer than more side constraints.

2 *PROBLEM FORMULATION*

The most familier application of GTP is the machine loading problem [12]. In that problem there are m types of machines which can produce n types of products. When machine i is used to produce product j it requires e_{ij} hours per unit and costs c_{ij} dollars per unit. It is assumed that during the planning period the available time in machine i is a_i hours, and the demand for product j is b_j. The problem is to determine x_{ij}, the amount of product j to be produced on machine i during the planning period, so that the required production is achieved without exceeding available machine hours, and also so that the total cost is minimized. In addition, there may be other constraints called side constraints which also have to be satisfied. Formulated as a linear program, the GTP is:

$$\text{Min } \{f = \Sigma_{i=1}^{m}\Sigma_{j=1}^{n} c_{ij}x_{ij} \}$$

$$\Sigma_{j=1}^{n} e_{ij}x_{ij} \le a_i \qquad \text{for } i=1,2,..,m$$

$$\Sigma_{i=1}^{m} x_{ij} \ge b_j \qquad \text{for } j=1,2,..,n$$

$$\Sigma_{i=1}^{m} \Sigma_{j=1}^{n} s_{kij}x_{ij} \le d_k \qquad \text{for } k=1,2,..,t.$$

$$x_{ij} \ge 0.$$

We call this problem P_{min}.

In this problem, the first set of m constraints makes certain that the available hours on each machine are not exceeded. The second set of n constraints requires that the the stated demands are met. The next set of t constraints are the extra or side constraints. In the above problem P_{min} it is assumed that e_{ij}, a_i, $b_j > 0$, and also that $c_{ij} \ge 0$. (In Section 4, we discuss the case in which some of the c_{ij} are negative).

The dual of the above problem is given by:

$$\text{Max}\{z= -\Sigma_{i=1}^{m} a_iu_i + \Sigma_{j=1}^{n} b_jv_j - \Sigma_{k=1}^{t} d_kw_k\}$$

$$\text{s.t.} \quad -e_{ij}u_i + v_j - \Sigma_{k=1}^{t} s_{kij}w_k \le c_{ij} \quad \text{for } i=1,2,..,m \text{ and } j=1,2,..,n$$

$$u_i,\ v_j,\ w_{ij} \ge 0.$$

We call this problem P_{max}.

For convenience, we also assume that, in the dual problem, a regularization constraint of the form:

$$\Sigma_{i=1}^{m} \quad u_i \leq M$$

is included in the constraint set as the mth constraint (M is a very large number).

3 *DESCRIPTION OF THE PIVOT AND PROBE ALGORITHM*

In order to make this paper self-contained, we present a shortened version of the Pivot and Probe Algorithm given in Sections 2 and 3 of [27]. The meanings of symbols used are to hold only in the present section.

Consider a linear programming problem stated in maximization form:

$$\text{Max } \{z = ex\}$$

$$\text{S.L } Ax \wedge b \qquad \text{(H)}$$

$$x * 0$$

where A is m x n, b is m x l, c is l x n, x is n x l and z is a scalar. We assume that the problem has been transformed so that b £ 0 which makes x = 0 a feasible solution. For convenience, we also assume that a regularization constraint of the form

$$r_{j} \; x_{j} <> M \qquad (1)$$

is included in the constraint set as the mth constraint where M is a very large number. Constraint (1) insures that the primal constraint set of II is bounded.

Next we define the following index sets:

$$I = (I, 2, 3\ldots\ldots, m\}$$

$$J = (I, 2, 3\ldots\ldots, n)$$

Using the above we define the index sets K° and K^1 of *candidate constraints of degrees 0 and* 7, see also [26], as the sets of indices of those constraints satisfying the following requirements:

$$K^\circ = 1m), \text{ m being the index of the regularization constraint} \qquad (2)$$

$$K^1 = U \{i \mid b./a..= \min b/.a_k. \text{ where } i, h \; G \; I, \; a. >0 \} \; UK^\circ \qquad (3)$$

In words, K^1 consists of the set of indices of all rows which could possibly be pivoted on at the first step of the simplex method together with the regularization constraint m. Geometrically, K^1 consists of the constraints which have an intercept on some coordinate axis which is closest to the origin together with constraint m. Note that K^1 will include.all constraints i for which $b_i = 0$.

In general we define the index set K^s of *candidate constraints of degree s* as:

$$K^s = U \{i \mid b_i/a_i \wedge \min b_h/a_{hj},\ a_{hj}>0 \text{ where } i,h \in IxljJ\ K^l \text{ for } t=1,..,s-1 \} \quad (4)$$

Geometrically, K^5 consists of the indices of those constraints which have an intercept on some coordinate axis which is the sth closest to the origin, and which have no other intercept that is ranked closer than s on any coordinate axis.

We now define index sets

$$I^s = t\gtrsim,\ K^h \quad (5)$$

$$I^N = I \setminus I^s \quad (6)$$

By means of these sets we define the linear programming problem II^s as follows:

$$\max\ ex$$

$$\text{s.L}\quad 2J_J\ a_{ij}x_j\ *\ b_i \text{ for } i \in I^s \quad \text{(ff)}$$

$$x_j \geq 0.$$

Because II^s contains fewer constraints than II, it follows that problem II^s is a relaxation of problem IL Because I^s includes the regularization constraint (1), Problem IT always has a solution if problem II does. Let x^D and w^D be primal and dual solutions to problem II^s which can be found by applying the simplex method to II^s, then w^D, when extended by adding 0 components for indices $i \in I^N$, is dual feasible for problem II; it also follows that $w°b$ is an upper bound to the optimal objective value z° of II. Of course, solution x^D may or may not be primal feasible for problem II.

Let x^P be any primal feasible (not necessarily basic) solution for problem II. We now want to define a *probe* in terms of the vectors x^D and x^P. A *probe* is the operation of finding the piercing points (if any) of the line segment between x^P and x^D and the constraints whose indices are in I^N; in particular we want to find the piercing point which is closest to x^P. If x is an arbitrary point on the line segment between x^P and x^D it can be written as:

$$x= (1-X)x^P + Xx^D \quad \text{for some } X \in [0,1] \quad (7)$$

Let $h \in I^N$; then the piercing point of the line segment (7) and constraint h is obtained by solving the equation

$$A_n x_{-} = (1 - \backslash_h) A_n / + X_n . A_n . x^{\circ} \tag{8}$$

for X^h. This gives

$$\backslash = \left(b_h - A_h x^P\right) / \left(A_h x^D - A_h x^P\right) \tag{9}$$

If x^D violates constraint h, that is, $A_n x^D > b\backslash$ then since x^p is primal feasible, it follows that X_h in (9) lies in the interval [0,1). Substituting this value of X_{u_n} into (8) gives the required piercing point

We shall say that constraint i is the *most violated constraint* if

$$X_i = \min X_h \text{ for } h \text{ € } H \tag{10}$$

where the set H is defined by

$$H = \{ h \text{ G } I^N \text{ I } A_h x^D > b_b \} \tag{11}$$

Substituting in the value of X_i from (10), the piercing point $\bar{x}$ of line segment (7) and constraint i is given by

$$\bar{x} = (1-X_i)x^p + X_i x^D \tag{12}$$

From (9) and (10) it follows that $\bar{x}$ is primal feasible for problem II; and it also follows that ^{cx} is a *lower bound* for the optimal value z° of n. Note also that x^p is determined by substituting into a single constraint, the most violated one, and hence is determined very accurately. If $X_i = 1$ in (10), Le., $A_i . x^D$ £ b^h for all h G I^N, then x^D is feasible for problem II, and there is no most violated constraint; hence x^D is optimal for II.

Note that, during the probe step, it is possible to probe to any previously known primal feasible point x^p from the dual feasible point x^D. Hence we keep a list, L, of primal feasible points which we wish to use for probing as the method proceeds. There are many ways to generate this list; we usually use the rule that L consists of the origin 0, and the most recent primal feasible solution. For each x in L we calculate its piercing point $\bar{x}$ with most violated constraint We then calculate the one whose objective value is largest, as follows:

$$Z = \max c\bar{x} \tag{13}$$

It follows that Z is a new lower bound to the optimal value of the linear program. If L contains the most recently found primal feasible solution, the new bound is no smaller than the previously found lower bound.

We now give a general description of the pivot and probe algorithm. A specialized implementation is discussed in Section 4.

PAPA (Pivot and Probe Algorithm!

1. (Initialization). Select the degree s; calculate V using (2)-(5); let $I^* = I^s$. Let IT be the linear program with constraints in I^*. Let the list of primal feasible solution be

   L = {0} where 0 is the 0 vector corresponding to the origin. Set LB = 0 and UB = oo be the initial lower and upper bounds to the optimal objective value.

2. Use the primal simplex method to solve problem II*; let x^D be its primal solution; let UB = cx^D.

3. Let V = 4>. For each x in L, probe to find the most violated constraint and piercing point $\bar{x}$; put the index of the most violated constraint in set V. Let x^p be found as in (13) as the piercing point giving the largest lower bound; let LB = cx^p. Update L.

4. If V = 4> go to 7. Otherwise go to 5.

5. Replace I* by I* U V.

6. Use the dual simplex method or the artificial variable method to solve IT; let x^D be its primal solution; let UB = cx^D. Go to 3.

7. Stop. The most recent x^D solution is optimal for problem n.

For some applications in which only an approximate optimal solution is needed, we can replace step 4 of the algorithm by step 4a below:

4a. If V = • or (UB - LB) £ E, go to *1.* Otherwise go to 5.)

Here E is the allowable error, and we decide to stop the computation whenever we find a primal feasible solution which is known to be within E of the optimum. Some computational experience with this rule will be discussed in Section 4. The way that L is updated in our current code is also discussed in Section 4.

4 SOLUTION OF GENERALIZED TRANSPORTATION PROBLEMS

We now describe how we solve a GTP by the Pivot and Probe Algorithm. The meanings of symbols in this section and all later sections is the same as those in Section 2.

If we return to problem P_{mm} in Section 2, we see that it has m x n variables and m + n +

t constraints; similarly P_{max} has m + n + t variables and m x n constraints. For example, if m = n = 100 and t = 10, then P_{min} is 210 x 10,000 while P_{max} is 10,000 x 210. In spite of the large number of constraints in P_{max} we will make our primal problem and use the fact that PAPA considers only a very small number of constraints at any one time. In fact the number of constraints in the relaxed linear program which is solved by PAPA will never exceed m + n + t, and usually considerably smaller.

We assumed in Section 2 that c_{ij} are non-negative. Hence a feasible solution to P_{max} is u_i = 0, v_j=0 and w_k = 0. All of the computational experience in Section 5 is based on the assumption of the non-negativity of c_{ij}.

In case $c_{ij} < 0$ for some pairs i and j we could use the following solution

$$u_i = -y^e u \quad \text{for} \quad c_{ij} < 0$$

$$u_i = 0 \quad \text{if} \quad c_{ij} \geq 0$$

and v_j=0 and w_k = 0

We have not, as yet, tested problems with $c_{ij} < 0$.

We define the supply-demand ratio as the ratio of the total available supply to the total available demand Thus, the supply-demand ratio, R will be given by

$$R = \sum_{i=1}^{m} a_i \Big/ \sum_{j=1}^{n} b_j \quad .$$

In the computer implementation we chose the degree of candidate constraints to be s = 1, which we had previously found in [27] to be a good choice. Also, as in [27], we let L consist of the two vectors 0 and X^p, where X^p was the most recently found primal solution. Each time a probe was made from the origin, the most violated constraint was saved, and also the second, third, ..., up to fifth or sixth most violated constraints were also saved and added to the relaxed problem. Also when a probe was made from X^p, the most recently found primal feasible solution, only the most violated constraint was added to the relaxed problem. Thus, at each probe step, as many as 5 or 6 new constraints were added to the relaxed problem.

We also found it necessary to drop constraints prior to each probe step to prevent the relaxed problem from becoming too large. The rule was to drop constraints in the relaxed

problem whose slacks were positive. It should be noted that basis reinversion was never needed for the solution of the problems reported on in this paper.

5 *COMPUTATIONAL RESULTS*

Computational results were obtained on a DEC VAX-11/780, using a FORTRAN 77 compiler. These results are summarized in Tables 1 through 7.

In Tables 1, 2 and 3, we solve dense problems (all the possible arcs exist) whereas in Tables 4 through 7, only sparse problems are solved. Also, in the former three tables, the coefficients e_{ij} are uniformly distributed between .1 and 3 as against between .5 and 1.5 in the latter four tables.

Table 1 gives the effect on solution time of varying supply-demand ratio, R. It can be seen that the problems become more and more difficult as this ratio approaches 1. However, with a fixed supply-demand ratio, it is more difficult to solve larger problems than smaller, and even more difficult if these problems have some side constraints. This conclusion is drawn from Table 2 where R is held constant at 2. The column 't=0' corresponds to problems with no side constraints and the column under the heading 't=5' to problems with 5 side constraints. In Table 3, it can be seen that for a fixed size (60x60) and fixed supply-demand ratio (R=1.5), the solution time goes up. For example, it took only half minute, on the average, to solve a pure GTP (line 1) as against more than 3 minutes in the presence of 10 side constraints (line 6). The last column in this table give the percentages of the side constraints which are binding at the optimum.

The dependence of solution time on the size of a GTP and the number of side constraints in a GTP, as seen above, holds true in case of sparse problems too. However, as evident in Table 4, it is much easier to solve a sparse problem than a dense one. By density of a problem, we mean the ratio of the constraints (arcs) in the problem to the total number of constraints possible. Thus, a 100x100 GTP can have 10,000 possible arcs. But a GTP of the same size with a density equal to .2 will have only about 2,000 arcs. As mentioned earlier, in this and the rest of the tables, the coefficients e_{ij} range between .5 and 1.5. The supply-demand ratio R in these four tables is fixed at 3.

Tables 5 and 6 show the effect on solution time of changing problem size with no side constraints (Table 5), and with 5 side constraints (Table 6). In both these tables, as also in table 7, the density of the problems solved was .1. It can be seen from Table 7 that like the dense problems, sparse problems too are harder to solve for a larger number of side constraints. Note that the number of pivots, as shown in some of the tables, is quite small

Number of Problems solved	Supply-demand Ratio, R	Solution time in Seconds	Number of Pivots
5	2.0	47	291
5	1.8	92	484
11	1.6	86	444
8	1.4	87	460

Table 1. The effect on solution time of varying supply-demand ratio R is shown here. Calculations were done on dense problems with m=n=60, side constraints t=4. Thus, in problem P_{max}, the number of variables was 124, and the number of constraints was 3600. Note that the solution time goes up rapidly as R approaches 1.

Problem Size	Number of Constraints	Time in Seconds t=0	t=5	Number of Pivots t=0	t=5
40x40	1600	7.2	18	103	226
60x60	3600	18.0	34	137	254
80x80	6400	52.6	117	214	397
100x100	10000	98.0	218	251	554

Table 2. The effect on solution time of changing problem size for cases of no side constraints (t=0) and five side constraints (t=5). Here the supply-demand ratio was held constant at R=2. Each solution time was computed as the average time of five randomly generated dense problems.

Problems Solved	# of Side Constraints	Time in Sees	% of side Constraints binding at Optimum	Number of Pivots
5	0	32	-----	214
5	2	65	70	295
5	4	121	80	596
5	6	125	70	581
5	8	153	81	702
2	10	190	70	845

Table 3. The effect on solution time of changing the number of side constraints. All P_{max} problems had 120+t variables and 3600 constraints. The supply-demand ratio was held constant at R = 1.50.

Problem Size	Problem Density	Number of Constraints	Time in Seconds	Number of Pivots
100x100	.1	1028	8.72	155
100x100	.3	2985	15.23	152
100x100	.5	5088	24.64	158
100X100	.7	6755	28.30	151

Table 4. The effect on solution time of changing problem density. There were no side constraints in P_{rain}. Here the supply-demand ratio was held constant at R=3. Each solution time was computed as the average time of seven randomly generated problems.

Problem Size	Number of Constraints	Solution Time in Seconds
60x60	388	2.45
100x100	1085	8.21
140x140	2032	23.21
180x180	3323	52.01

Table 5. The effect on solution time of changing problem size for case of five side constraints (t=0) in P_{min}. Here the supply-demand ratio was held constant at R=3. The density of the problems solved was 0.1. Each solution time was computed as the average time of seven randomly generated problems.

Problem Size	Number of Constraints	Solution Time in Seconds
40X40	175	3.07
60X60	391 '	9.84
80X80	651	24.25
100X100	1029	45.04
120x120	1508	87.50

Table 6. The effect on solution time of changing problem size for case of five side constraints (t=5) in P_{min}. Here the supply-demand ratio was held constant at R=3. The density of the problems solved was 0.1. Each solution time was computed as the average time of seven randomly generated problems.

Problem Size	# of Side Constraints	Number of Constraints	Time in Seconds	Number of Pivots
70x70	0	518	3.08	103
70X70	2	499	6.62	137
70X70	4	502	13.46	185
70X70	6	537	25.50	217
70X70	8	525	35.54	288

Table 7. The effect on solution time of changing the number of side constraints in P_{rain}. About 70-90% of the side constraints were binding at the optimum. Here the supply-demand ratio was held constant at R=3. Each solution time was computed as the average time of seven randomly generated problems each having density 0.1.

6 *CONCLUSION*

It is probable that the Pivot and probe Algorithm is slower than existing network codes in solving a GTP without side constraints. However, we know of no network code that can solve a GTP with more than 1 or 2 side constraints. Recently, Gupta and McBride in [20] have developed a specialized linear programming code for solving a GTP which has an arbitrary number of side constraints (and also side variables). We have found our version of the simplex method to be very slow as compared to PAPA for solving constrained GTFs. Further work along these lines is in progress.

References

1. Balachandran, V., and Thompson, G.L., "An Operator Theory of Parametric Programming for the Generalized Transportation Problem: I. Basic Theory", *Naval Research Logistics Quarterly,* 22, 1, 1975, 79-100.

2. Balachandran, V., and Thompson, G.L., "An Operator Theory of Parametric Programming for the Generalized Transportation Problem: II. Rim, Cost and Bound Operators", *Naval Research Logistics Quarterly,* 22, 1, 1975, ldl-126.

3. Balachandran, V., and Thompson, G.L., "An Operator Theory of Parametric Programming for the Generalized Transportation Problem: III. Weight Operators", *Naval Research Logistics Quarterly,* 22, 2, 1975, 297-316.

4. Balachandran, V., and Thompson, G.L., "An Operator Theory of Parametric Programming for the Generalized Transportation Problem: IV. Global Operators", *Naval Research Logistics Quarterly,* 22, 2, 1975, 317-340.

5. Balas, E. and Ivanescu, P.L., "On the Generalized Transportation Problem", *Management Science,* 11, 1965, 188-202.

6. Balas, E», "The Dual Method for the Generalized Transportation Problem", *Management Science,* 12, 7, 1966.

7. Bradley, G.H., Brown, *G.G.,* and Graves, G.W., "Design and Implementation of Large-Scale Primal Transshipment Algorithms", *Management Science,* 24, 1, 1977, 1-34.

8. Brown, G. and Mcbride, R., "Solving Generalized Networks," Presented at the Detroit ORSA/TIMS Meeting, April 1981

9. Charnes, A. and Cooper, W.W., *Management Models and Industrial Applications of Linear Programing,* Vols. I & II. New York: John Wiley and Sons, 1961.

10. Chen, S. and Saigal, R.,"A Primal Algorithm for Solving A Capacitated Network Flow Problem with Additional Linear Constraints", *Networks,* 7, 1, 1977, 59-79.

11. Eisemann, K., "The Generalized Stepping-Stone Method for the Machine Loading Model", *Management Science,* 11, 1, 1964.

12 Eisemann, K. and Lourie, J.R., "The Machine Loading Problem", IBM 704 Program, IBML-1, IBM Application Library, New York, 1959.

13. Fergusan, A.R., and Dantzig, G.B., "The Allocation of Aircrafts to Routes - An Example of Linear Programming Under Uncertain Demand", *Management Science,* 3, 1, 1956.

14. Glover, F., Hultz, J., and Klingman, D., "Improved Computer Based Planning

Techniques", Research Report CCS 283, Center of Cybernetic Studies, The University of Texas, Austin, Texas, 1977.

15. Glover, F., Hultz, J., Klingman, D., and Stutz, J., "Generalized Networks: A Fundamental Computer-Based Planning Tool", *Management Science,* 24, 12, 1978, 277-297.

16. Glover, F., Karney, D. and Klingman, D., "Implementation and Computational Comparisons of Primal, Dual and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems", *Networks,* 4, 3, 1974, 191-212.

17. Glover, F., Karney, D., Klingman, D., and Russell, R., "Solving Singly Constrained Transshipment Problems", *Transportation Science,* 12, 4, 1978, 277-297.

18. Glover, F. and Klingman, D., "A Note on Computational Simplification in Solving Generalized Transportation Problem", *Transportation Science, 1,* 4, 1973, 351-361.

19. Glover, F. and Klingman, D., "New Advances in the Solution of Large-Scale Network and Network Related Problems", Technical Report CCS 177, Center of Cybernetic Studies, the University of Texas, Austin, Texas, 1974.

20. Gupta, A. and McBride, R., "Solving Embedded Generalized Network Problems", Presented at the ORSA/TIMS Meeting in Detroit, April 1982.

21. Helgason, R.V. and Kennington, J.L., *Algorithms for Network Programming,* Publisher. John Wiley & Sons, New York, 1980.

22. Hultz, J., and Klingman, D., "Solving Singularly Constrained Generalized Network Problems", *Applied Mathematics and Optimization,* 4, 1978, 103-119.

23. Klingman, D., and Russell, R., "On Solving Constrained Transportation Problems", *Operations Research,* 23, 1, 1975, 91-107.

24. Lourie, J.R., "Topology and Computation of Generalized Transportation Problem', *Management Science,* 11, 1965, 177-187.

25. Mcbride, R.D., "Solving Generalized Network Problems with Side Constraints", Presented to the CORS/ORSA/TIMS Meeting, Toronto, May 1981.

26. Sethi, A.P. and Thompson, G.L., "The Non-candidate Constraint Method for Reducing the Size of a Linear Program", in *Redundancy in Mathematical Programming* edited by Karwan, Lotfi, Telgen, and Zionts; Springer-Verlag, 1983.

27. Sethi, A.P., and Thompson, G.L., "The Pivot and Probe Algorithm for Solving A Linear Programming Problem", Working Paper #36, April 1982, GSIA, Carnegie-Mellon University, Pittsburgh; to appear in *Mathematical Programming.*

28. Srinivasan. V. and Thompson, G.L., "Accelerated Algorithms for Labelling And Re-Labelling of Trees, with Applications to Distribution Problems", *Journal of the Association for Computing Machinery,* 19, 4, 1972, 712-726.

29. Srinivasan, V. and Thompson, G.L., "Benefit-Cost Analysis of Coding Techniques For the Primal Transportation Algorithm", ***Journal of the Association for Computing Machinery,*** **20, 2, 1973, 194-213.**