

AD-763 381

THE SINGULARLY CONSTRAINED GENERALIZED NETWORK PROBLEM

Fred Glover, et al

Texas University

Prepared for:

Office of Naval Research

February 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

**Best
Available
Copy**

THE SINGULARLY CONSTRAINED
GENERALIZED NETWORK PROBLEM

by

Fred Glover*

D. Klingman**

**CENTER FOR
CYBERNETIC
STUDIES**

The University of Texas
Austin, Texas 78712

Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U.S. Department of Commerce
Springfield, VA 22151



THE SINGULARLY CONSTRAINED
GENERALIZED NETWORK PROBLEM

by

Fred Glover*

D. Klingman**

* Professor of Management Science at the University of Colorado,
Boulder, Colorado

** Associate Professor of Operations Research at the University of Texas
Austin, Texas

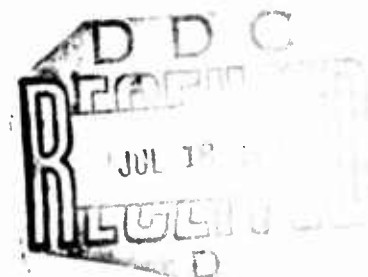
February 1973

This research was partly supported by a grant from the Farah Foundation
and by ONR Contracts N00014-67-A-0126-0008 and N00014-67-A-0126-0009
with the Center for Cybernetic Studies, The University of Texas. Reproduction
in whole or in part is permitted for any purpose of the United States
Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 512
The University of Texas
Austin, Texas 78712

///



DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Center for Cybernetic Studies University of Texas at Austin		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE The Singularly Constrained Generalized Network Problem			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name) F. Glover D. Klingman			
6. REPORT DATE February 1973		7a. TOTAL NO. OF PAGES 16 / 9	7b. NO. OF REFS 15
6a. CONTRACT OR GRANT NO. NR-047-021		9a. ORIGINATOR'S REPORT NUMBER(S) Center for Cybernetic Studies Research Report No. 110	
b. PROJECT NO. N00014-67-A-0p26-0008		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c. N00014-67-A-0126-0009			
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Office of Naval Research (Code 434) Washington, D.C.	
13. ABSTRACT <p>This paper presents a computationally efficient method for solving generalized network problems with an additional linear constraint. The method is basically the primal simplex method specialized to exploit the topological structure of the problem. The method is similar to the specialization of Charnes and Cooper's Double Reverse Method by Meier, and Klingman and Russell for constrained pure network problems. It couples the augmented predecessor index method with a double pricing procedure to yield an "inverse compactification" which reduces the arithmetic calculations required in pivoting. We also show how to simplify and accelerate the steps of updating costs and finding basis representations by taking advantage of the quasi-triangularity of a basis.</p>			

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Generalized Networks						
Weighted Networks						
Networks with gains						
Constrained Networks						
Linear Programming						

//

Unclassified

Security Classification

Abstract

This paper presents a computationally efficient method for solving generalized network problems with an additional linear constraint. The method is basically the primal simplex method specialized to exploit the topological structure of the problem. The method is similar to the specialization of Charnes and Cooper's Double Reverse Method by Meier, and Klingman and Russell for constrained pure network problems. It couples the augmented predecessor index method with a double pricing procedure to yield an "inverse compactification" which reduces the arithmetic calculations required in pivoting. We also show how to simplify and accelerate the steps of updating costs and finding basis representations by taking advantage of the quasi-triangularity of a basis.

1. INTRODUCTION

In this paper we present a computationally efficient method for solving generalized network problems with an extra linear constraint. Such linear models occur frequently in network applications. For example, a variety of scheduling models, such as constrained machine scheduling models [2, 4, 11], copper refining process models [8], sewage treatment models [12], and power transformer inventory models [14], fall into this class of problems. Also, cash flow models [2, 15] and gas blending models [2, 3] are of this type. Motivation for the development of a computationally efficient solution method is reinforced by the potential use of such a method to solve subproblems generated by branch-and-bound and cutting plane procedures for integer generalized network problems.

The proposed solution method is basically the primal simplex method specialized to take advantage of the augmented predecessor index method [5] used in codifying generalized network solution algorithms [1, 4, 6, 11]. Our approach is similar to that used by Maier [13], and Klingman and Russell [9, 10] for solving constrained pure network problems. In particular, it couples the augmented predecessor index method with a double pricing procedure (similar to the Poly- w technique of Charnes and Cooper [2]) to yield an "inverse compactification" which significantly reduces the arithmetic calculations required in pivoting. We show also how to simplify and accelerate the steps of updating costs and finding basis representations by exploiting the quasi-triangularity of a basis.

2.0 THE GENERALIZED NETWORK PROBLEM

The generalized network problem with an extra constraint may be defined

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{(i,j) \in A} a_{p,ij} x_{ij} = b_p, \quad p \in N \quad (2)$$

$$0 \leq x_{ij} \leq u_{ij}, \quad (i,j) \in A \quad (3)$$

$$\sum_{(i,j) \in A} d_{ij} x_{ij} = d_0 \quad (4)$$

where A is the set of arcs and N is the set of nodes for the network. Each arc (i,j) , $i \neq j$, has a nonzero coefficient in exactly two of the node equations (2), i.e., the two equations corresponding to the arc's endpoints. Specifically, $a_{p,ij} \neq 0$ only if $p = i$ or $p = j$. In an ordinary network $a_{i,ij} = -1$ and $a_{j,ij} = 1$, but in a generalized network $a_{i,ij}$ and $a_{j,ij}$ can be any two nonzero quantities. Typically, however, $a_{i,ij}$ is assumed to be -1 and $a_{j,ij}$ is assumed to be positive, in which case $a_{j,ij}$ is called the "multiplier" of the arc directed from node i to node j . This multiplier can be thought of as a factor which magnifies or attenuates the flow x_{ij} across the arc, according to whether $a_{j,ij}$ is greater or less than one.

A generalized network can also contain arcs which are "self-loops," leading from a node back to itself. That is, for some nodes i , there may exist "arcs" (i,i) in A . In this case, $a_{p,ii} \neq 0$ only if $p = i$. Such self-loops are customarily used to introduce slack variables into the problem, (e.g., to change inequalities into equations and have been called slack loops [3, p. 413-424]). In addition, there may be a slack variable for equation (4). If so, we will denote this variable for convenience as x_{00} , creating a "special" arc $(0,0) \in A$ which has no endpoints; i.e., $0 \notin N$. Thus, $a_{0,ij}$ does not exist for any ij (i.e., $a_{p,00} = 0$ for all $p \in N$). These conventions make it possible to accommodate all problem variables by means of a uniform notation, simplifying the statement

of a number of results. (Prescriptions involving $a_{r,rs}$ and $a_{s,rs}$ when $r=s=0$, for example, are to be ignored, whereas prescriptions involving c_{00} or d_{00} are to be followed as specified.)

The quantities b_p of the node equations represent the supplies and demands at the nodes, where $b_p > 0$ is interpreted as a demand, and $b_p < 0$ is interpreted as a supply.

3.0 STRUCTURE OF BASIC SOLUTIONS

The structure of a basis for problem (1) - (4) is related in a simple fashion to the structure of a basis for an ordinary generalized network problem ((1)-(3)). We will make several preliminary observations concerning this structure and the implications of this structure for identifying variables and marginal cost values given a basis. These observations are relatively direct specializations and/or extensions of well-known results about linear programming problems and generalized network problems and will be presented without proofs [1, 2, 3, 4, 6, 9, 10, 13, 14]. In section 4 we will make use of these foundations to develop basis updating procedures that provide an efficient specialization of the primal simplex method to the current problem.

Our first observation characterizes the topological structure of the basis for problem (1) - (4).

Remark 1: A basis for (1)-(4) consists of a spanning collection of disjoint subnetworks, one of them a tree with 2 additional arcs, and each of the remaining subnetworks a tree with one additional arc.

(It should be noted that the "nodeless" arc (0,0) corresponding to a slack variable for (4) qualifies as a "tree with 2 additional arcs," since the number of arcs in a tree is one

less than the number of nodes, whereas the arc $(0,0)$ itself, having no endpoints, contains two arcs more than a "tree" on zero nodes.)

We shall refer to the subnetworks of the basis identified in Remark 1 as quasi-trees, and refer to the particular quasi-tree with 2 additional arcs as the distinguished quasi-tree. The remaining subnetworks of the basis will be called ordinary quasi-trees.

By pruning a quasi-tree we will mean removing those arcs which connect to nodes that have exactly one arc incident to them, and repeating the process in the connected network that results, until all nodes remaining have at least 2 arcs incident to them.

Remark 2: Pruning an ordinary quasi-tree leaves a single loop (with every node of order 2) and pruning the distinguished quasi-tree leaves either two loops connected by a chain or two "overlapping" loops composed of 3 arc-disjoint chains which share common endpoints (in each case, with two nodes of order 3 and all remaining nodes of order 2). [An exception occurs if the distinguished tree is the arc $(0,0)$ in which case pruning leaves the quasi-tree just as it was - with 1 arc and 0 nodes.]

We will call an arc of the distinguished quasi-tree removable if its elimination leaves an ordinary quasi-tree.

Remark 3: An arc is removable if and only if it is contained in the distinguished quasi-tree after pruning.

The network that results from the basis network upon eliminating a re-

movable arc will be called a reduced basis. Thus, a reduced basis is a collection of ordinary quasi-trees, and qualifies as a basis for the ordinary generalized network problem (in which equation (4) is absent).

Relative to a specified removable arc, we shall define the reduced equation (4)

$$\sum_{(i,j) \in N} f_{ij} x_{ij} = f_0 \quad (5)$$

to be the equation that results after "pricing out" the original equation (4) with the set of arcs of the reduced basis. That is the coefficients d_{ij} of equation (4) are treated as though they are objective function coefficients and the "pricing out" identifies

$$f_{ij} = d_{ij} - \pi_i a_{i,ij} - \pi_j a_{j,ij}$$

where π_i and π_j are the node multipliers determined by the procedures for the ordinary generalized network problem such that $f_{ij} = 0$ for all arcs of the reduced basis (see, e.g. [6]). The quantity f_0 equals $d_0 - \sum_{p \in N} \pi_p b_p$. To price out the objective function (1) (i.e., to determine the marginal costs of the arcs) for the problem (1)-(4) involves two steps. First, (1) is priced-out with the reduced basis, to yield "updated objective function coefficients" c_{ij}^* given by

$$c_{ij}^* = c_{ij} - \pi_i^* a_{i,ij} - \pi_j^* a_{j,ij} \quad (6)$$

where the node potentials π_i^* and π_j^* are determined as in [6] so that $c_{ij}^* = 0$ for all (i,j) in the reduced basis. The second step is then achieved as indicated in the following remark.

Remark 4: Let (r,s) denote the removable arc and define $\theta = c_{rs}^* / f_{rs}$. Then the updated objective function coefficients relative to the full basis for (1) - (4) (i.e., the basis that includes the removable arc) are given by

$$c_{ij}' = c_{ij}^* - \theta f_{ij}, \quad (i,j) \in A.$$

The coefficients of (5) play a role not only in identifying marginal costs for the arcs, but also in determining the basic solution, as seen to follow.

Remark 5: The values of the variables in the basic solution to equation (2) and (4) are the same as for equations (2) and (5). Moreover,

the value of the variable x_{rs} (corresponding to the removable arc (r,s)) is f_0 / f_{rs} , and the values of the other basic variables are those for the ordinary generalized network problem (solving (2) relative to the reduced basis), with b_r and b_s replaced by $b_r - a_{r,rs} (f_0/f_{rs})$, and $b_s - a_{s,rs} (f_0/f_{rs})$, respectively.

4.0 UPDATING THE BASIS

The preliminary remarks of the preceding section may be applied in direct form to determine the values of the variables and the node potentials (dual evaluators) for a starting basis for problem (1)-(4). At subsequent iterations of a primal simplex algorithm, however, the values of the variables and node potentials are determined more conveniently and efficiently by reference to the values in the preceding iteration. It is this problem which we now address.

By the standard rules of the primal simplex method, the "incoming variable" (i.e., the variable which is chosen to enter the basis during the basis exchange step) is selected to be a nonbasic variable with a negative marginal cost in the updated objective function. Having identified such a variable, which we will hereafter designate x_{uv} , it is necessary to identify its basis representation.

By definition, the basis representation of (u,v) is the "basic solution" to the equations (2) and (4) with d_0 replaced by d_{uv} , b_u replaced by $a_{u,uv}$, b_v replaced by $a_{v,uv}$, and all other $b_p, p \in N$ replaced by 0. Thus, strictly speaking, Remark 5 can be applied to generate this representation. However, the basis structure and the predominance of the b_p 's that are replaced by 0 makes it possible to simplify

the calculations to a notable extent. To accomplish this in a convenient way, and to facilitate the calculations involved in other updating operations of the basis exchange step, we will organize our results around the use of the augmented predecessor indexing (API) method for generalized network problems [5], demonstrating how to take advantage of the API method in the present context.

Thus, following [5], we specify that the arcs of the initial reduced basis are given a rooted-loop orientation; i.e., in each quasi-tree of this basis, a predecessor indexing is assigned that identifies each loop node as the predecessor of all other nodes, including itself. Each tree that results by suppressing the loop arcs is oriented as an arborescence which is rooted at its respective loop node. In each quasi-tree of the reduced basis, so oriented, the backward path P_i from a given node i consists of the succession of nodes and arcs formed by starting at node i and proceeding from predecessor to predecessor until some node is intersected a second time. Thus, the backward path from any node always contains all loop arcs without duplication and duplicates only the node at which the loop is entered. The desired basis representation for the incoming arc (u,v) , which in turn makes it possible to determine the outgoing arc, is given by the following remark.

Remark 6: The arcs which receive a nonzero weight in the full basis representation of the incoming arc (u,v) are contained in the backward paths P_u and P_v from nodes u and v , and if $f_{uv} \neq 0$, also include the removable arc (r,s) together with the arcs contained in the backward paths P_r and P_s from nodes r and s .

Proof: The proof of this remark follows the reasoning of [5,6,11] noting that if $f_{uv} \neq 0$, then arc (r,s) must assume a nonzero weight by Remark 5 (with f_0 replaced by f_{uv} , applying Remark 5 to definition of the basis representation). The effect of imparting a nonzero weight to (r,s) (when $f_{uv} \neq 0$) in turn requires an assignment of weights to arcs of P_r and P_s (if these paths exist; i.e., unless

$r=s=0$) to resolve "imbalance"-or node requirements-thus created at nodes r and s .

It should be noted that the paths P_u, P_v, P_r and P_s of Remark 6 may intersect (or not) in a variety of ways. In fact, because u and v may either or both duplicate the nodes r and s , the number of distinct nodes (and backward paths) involved may be anywhere from 2 to 4.

For Remark 6 to be useful, we need to specify how to determine the basis representation from the indicated backward paths. This is accomplished by reference to Remark 5 and the "pruning" procedure characterized in Section 3 restricted to P_u, P_v, P_r and P_s . Note that to prune the network which consists of these paths (hence which consists of portions of one to four quasi-trees of the reduced basis), the nodes to be examined first by the pruning process are u, v, r and s , removing their attached arcs, proceeding then to the predecessors of these nodes on the backward paths, and so forth. Thus, the sequence of steps that accomplish the pruning are already conveniently "pre-programmed" by the sequences of the backward paths themselves, subject only to the qualification that no arc may be removed which succeeds a node at which two or more paths intersect until the "previous" arcs on all of the interacting paths have been removed.

In the construction of the basis representation, weights are assigned to the arcs in the order in which they are removed by this process. At the time a given arc (i, j) is dropped, it is the only arc incident to one of its endpoints (say node i , for example), and thus the weight w_{ij} to be assigned to the arc is uniquely determined by the node requirement at this node. Specifically if node i has a requirement of r_i , then the weight attached to arc (i, j) is given by $w_{ij} = r_i / a_{i,ij}$. The assignment of the weight w_{ij} to the arc not only satisfies the requirement at node i , of course, but also transmits a requirement $-w_{ij} a_{j,ij}$ to node j , which is added to whatever other requirement has thus far accumulated at j . (If the arc incident to node i is (i, j) the index ij is replaced by ji .) In this fashion, the pruning process automatically assigns weights to successive arcs of P_u, P_v, P_r, P_s , given an appropriate set of node requirements

at the initial nodes of these paths. Ultimately the pruning process applied to the network of P_u, P_v, P_r, P_s , leaves only a loop or set of loops, some of whose nodes have received requirements transmitted by arcs that lead into them along the backward paths. The assignment of weights to the loop arcs to meet these requirements is then determined by the computations prescribed in [6,14].

The pruning approach just outlined is the standard way for exploiting the predecessor indexing scheme, amended slightly to accommodate the more involved structure of the basis for the current problem. We complete this prescription for assigning weights to the basic arcs by indicating the node requirements that are attached to nodes u, v, r and s to initiate the procedure.

Remark 7: The requirements at nodes u, v, r and s for determining the basis representation of arc (u,v) are respectively $a_{u,uv}, a_{v,uv}, -a_{r,rs} f_{uv}/f_{rs}$ and $-a_{s,rs} f_{uv}/f_{rs}$, except that if node r or s corresponds to node u or v , then the requirements for such corresponding nodes are obtained by summing the individual requirements indicated for these nodes.

Proof: The specified requirements are a consequence of Remark 5 applied to the definition of the basis representation replacing f_o, b_u and b_v by $f_{uv}, a_{u,uv}$ and $a_{v,uv}$, and replacing b_r and b_s by 0, unless r or s corresponds to u or v , in which case b_r or b_s takes instead the value indicated for b_u or b_v .

Utilizing the basis representation determined for the incoming arc (u,v) by Remark 7 and its preceding discussion, the outgoing arc which is determined by the primal simplex method is identified by the customary rules, which are detailed in the context of the generalized network problem in [6].

The statement of these rules remains as in [6], provided allowance is made for the fact that the removable arc (r,s) is included in the basis representation with a weight of f_{uv}/f_{rs} .

The final goal in specifying an efficient procedure for exploiting the problem (1)-(4) is to determine an appropriate scheme for updating the values of the variables, the node potentials and the rooted loop orientation for the new basis that results from the basis exchange step. Updating the values of the variables is actually quite simple, involving nothing beyond the rules already specified in [6]. Determining the new rooted-loop orientation is more crucial, providing the foundation for generating new node potentials. A central consideration affecting the new rooted-loop orientation is the influence of the removable arc, and the possibility that this arc may have to change its identity. We will show that it is possible to accommodate this consideration in a particularly simple way, integrating a required change of identity of (r,s) with the other changes induced by the basis exchange, resulting in a minimal amount of updating effort - indeed scarcely more than required for the ordinary generalized network problem. The result that makes this possible is the following.

Remark 8: The removable arc is required to change its identity if and only if the outgoing arc does not lie on either of the two backward paths P_u and P_v . Moreover, whenever such a change of identity is necessary, the role of the new removable arc may be taken by the incoming arc (u,v) .

Proof: If neither P_u and P_v contains the outgoing arc, the addition of (u,v) has created a quasi-tree containing too many arcs to be contained in the reduced basis. This follows from the fact that the assertion is true for the "stripped" quasi-tree consisting of P_u , P_v and (u,v) , and thus must be true for the full quasi-tree since the latter is created by attaching trees which

add as many arcs as nodes (since the roots of these trees already lie in P_u and P_v). Thus the full quasi-tree is the new distinguished quasi-tree, and some arc in this pruned segment of this quasi-tree must assume the role of the removable arc. But in fact the pruned segment is precisely the subnetwork composed of P_u , P_v and (u,v) , permitting (u,v) to be the arc specified. On the other hand, if the outgoing arc is contained in P_u or P_v , then the changes in the reduced basis are those for the ordinary generalized network problem, producing a new reduced basis which consists of ordinary quasi-trees. Thus, in this case the removable arc need not change its identity.

The chief significance of Remark 8, by means of which it is possible to give a simple prescription for updating the rooted-loop orientation, is made evident in the following result.

Remark 9: If arc (r,s) is the outgoing arc, then no change in orientation is required. Otherwise, by substituting (r,s) for (u,v) in the role of the incoming arc whenever (u,v) takes the role of the new removable arc, the updating of the rooted-loop orientation is accomplished by the same rules that apply to the ordinary generalized network problem.

Proof: Provided (r,s) is not the outgoing arc; whenever (u,v) becomes the new removable arc, (r,s) actually becomes the "incoming arc" relative to the reduced basis, in accordance with the changes specified in Remark 8.

The application of Remark 9, as indicated in its statement, simply reduces to a direct application of the rules for the API method given in [5]. It follows that the updating of node potentials likewise reduces precisely to the

procedure indicated in [6] - a somewhat unexpected but pleasantly satisfying result. In the present setting, of course, the updating of these potentials is applied simultaneously to the coefficients of (1) and (4), giving rise to the coefficients f_{ij} and c_{ij}^* from which the final updated form of (1) is determined by Remark 4.

REFERENCES

- [1] Balas, Egon and P. L. Ivanescu (Hammer), "On the Generalized Transportation Problem," Management Science 11, 1(1964), 188-202.
- [2] Charnes, A. and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Volumes I and II. New York: John Wiley and Sons, Inc., 1961.
- [3] Dantzig, George, Linear Programming and Extensions. Princeton, N. J.: Princeton University Press, 1963.
- [4] Eisemann, Kurt, "The Generalized Stepping-Stone Method for the Machine Loading Model," Management Science 11, 1(1964), 154-177.
- [5] Glover, Fred, D. Klingman, and J. Stutz, "Extensions of the Augmented Predecessor Index Method To Generalized Network Problems," Research Report CS 124, Center for Cybernetic Studies, BEB-512, University of Texas, 1973.
- [6] Glover, F. and D. Klingman, "A Note On Computational Simplifications in Solving Generalized Transportation Problems " To appear in Transportation Science.
- [7] Jewell, W. S. "Optimal Flow Through Networks with Gains," Operations Research 10, 4(1962), 476-499.
- [8] Kim, Y. C., "An Optimal Computational Approach To The Analysis of a Generalized Network of Copper Refining Process," Department of Mining and Geological Engineering, University of Arizona, 1972.
- [9] Klingman, D. and R. Russell, "On Solving Constrained Transportation Problems," Research Report CS 91, Center for Cybernetic Studies, BEB-512, University of Texas, 1972.

- [10] Klingman, D. and R. Russell, "On Solving Singularly Constrained Transportation Problems," Research Report CS 95, Center for Cybernetic Studies, BEB-512, University of Texas, 1973.
- [11] Lourie, Janice, "Topology and Computation of the Generalized Transportation Problem," Management Science 11, 1 (1964), 177-187.
- [12] Lynn, W. R. and A. Charnes, "The Dyadic Computation of a Generalized Network Model of a Sewage Treatment System," ONR Research Memorandum No. 19, Northwestern University, Evanston, Illinois, October 1959.
- [13] Maier, Steven, "A Multi-Item Network Flow Model With Capacity Constraints," technical report No. 69-14, Operations Research House, Stanford University, Stanford, California.
- [14] Maurras, Jean Francois, "Optimization of the Flow Through Network with Gains," Mathematical Programming 3, (1972) 135-144.
- [15] Srinivasan, V. "A Transshipment Model for Cash Management Decisions," Management Sciences, Research Report No. 243, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania.