Title
A Lower Bound for Uncapacitated, Multicommodity Fixed Charge Network Design Problems

Authors
Lamar, Bruce W.
Sheffi, Yosef
Powell, Warren B.

Publication Date
1987-08-01

A Lower Bound for Uncapacitated, Multicommodity Fixed Charged Network Design Problems

UCI-ITS-WP-87-4

Bruce W. Lamar [1]
Yosef Sheffi [2]
Warren B. Powell [3]

[1] Graduate School of Management and
Institute of Transportation Studies
University of California, Irvine

[2] Department of Civil Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

[3] Department of Civil Engineering
Princeton University, Princeton, New Jersey

August 1987

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
http://www.its.uci.edu

ABSTRACT

Network design problems concern flows over networks in which a fixed charge must be incurred before an arc becomes available for use. The uncapacitated, multicommodity network design problem is modeled with (i) aggregate, and (ii) disaggregate "forcing" constraints. [Forcing constraints ensure logical relationships between the fixed charge-related and the flow-related decision variables.] A new lower bound for this problem —referred to as the capacity improvement (CI) bound—is presented; and an efficient implementation scheme (using shortest path and linearized knapsack programs) is described. A key feature of the CI lower bound is that based on the LP relaxation of the aggregate version of the problem. A numerical example illustrates that the CI lower bound (i) can be as tight as the disaggregate LP relaxation, and (ii) can converge to the optimal objective function value of the IP formulation.

INTRODUCTION

Fixed charge network design problems concern flows in networks in which a quantum of cost must be incurred before an arc becomes available for use. This problem structure, reflecting the cost tradeoffs between the procurement of and operation on networks links, has been used to model a wide variety of logistical problems including multicommodity transshipment and distribution, facility location, highway planning, and computer/communications networking (Magnanti and Wong (1984)).

This paper focuses on a new lower bound for the class of multicommodity network design problems whose arc cost function consists of a fixed charge and a linear routing cost. Since the routing cost is linear, these problems are uncapacitated. A distinctive feature of this lower bound is that it is based on an aggregate problem formulation; that is, aggregate (rather than disaggregate) constraints are used to enforce logical relationships among the decision variables. Compared to their disaggregate counterparts, aggregate formulations are considered "weaker" in the sense that the LP relaxation of aggregate formulations is generally looser than that of disaggregate ones. This paper presents theoretical and computational results, however, which show that, in some circumstances, this new lower bound can be as tight as the disaggregate LP relaxation. Thus, strong bounds <u>can</u> be derived from aggregate formulations.

The paper is organized into four sections. Section 1 formulates the fixed charge network design problem as an aggregate integer programming problem and compares this program to alternative disaggregate formulations. Section 2 presents a new lower bound to the integer program and describes an efficient algorithm for computing it. Section 3, using a set of numerical examples, illustrates the use of this algorithm. Finally, Section 4 concludes the paper.

1. MODEL FORMULATION

The focus of this section is on problem formulation. It includes three parts. The first part defines notation and formulates the uncapacitated, multicommodity, fixed charge network design problem as an aggregate integer program. The second part shows that the LP relaxation of this problem has a very simple structure. The third part introduces the more common disaggregate formulation of the problem considered here and compares it to our aggregate formulation.

• Integer Programming Formulation

The following notation is used to define the problem. Let

$\underset{\sim}{N}$ = Set of nodes with generic element n

$\underset{\sim}{A}$ = Set of directed arcs with generic element a

$\underset{\sim}{K}$ = Set of commodities with generic element k

For each arc $a \in \underset{\sim}{A}$, the "tail" and "head" nodes incident to arc a are denoted by I(a) and J(a), respectively. Also, for each node $n \in \underset{\sim}{N}$, let $\underset{\sim}{A}_n$ denote the set of arcs whose tail node is n and let $\underset{\sim}{B}_n$ denote the set of arcs whose head node is n. That is,

$\underset{\sim}{A}_n = \{ a : a \in \underset{\sim}{A} \text{ and } I(a)=n \}$

$\underset{\sim}{B}_n = \{ a : a \in \underset{\sim}{A} \text{ and } J(a)=n \}$

For each commodity $k \in \underset{\sim}{K}$, let O(k) and D(k) denote, respectively, the "origin" and "destination" nodes for commodity k; and let d_k

denote the quantity of commodity k supplied at node O(k) and demanded at node D(k). It is assumed that d_k is nonnegative.

The fixed charge for arc a is denoted by f_a and the routing cost (i.e., cost per unit flow) for arc a and commodity k is denoted by $c_{a,k}$. It is assumed that these coefficients are nonnegative. In addition, a parameter used in the aggregate problem formulation is u_a^{max}. This parameter denotes the maximum flow coefficient for arc a and can be taken as the sum $\sum_k d_k$. Note that this coefficient is not intended to capacitate the problem; rather, it is used in the problem formulation to enforce logical relationships among the decision variables.

The decision variables in the problem designate which arcs are selected in the network design and how much flow is carried on each of the selected arcs. For each arc a, let $y_a = 1$ if arc a is selected to be in the network and let $y_a = 0$ otherwise. Also, let $x_{a,k}$ denote the flow of commodity k carried on arc a and let x_a denote the sum of the commodity flows carried on arc a. The $\{x_{a,k}\}$ and $\{x_a\}$ decision variables are referred to, respectively, as the commodity-specific and the aggregate arc flows.

As part of the problem formulation, let $\underset{\sim}{X}$ denote the set of (aggregate) arc flow vectors $\underline{x} = (\ldots,x_a,\ldots)$ that conform to the following multicommodity flow balance equations:

$$\sum_{a \in \underset{\sim}{A}_n} x_{a,k} - \sum_{a \in \underset{\sim}{B}_n} x_{a,k} = \begin{Bmatrix} d_k & \text{if } n=O(k) \\ -d_k & \text{if } n=D(k) \\ 0 & \text{otherwise} \end{Bmatrix} \qquad \forall\ n,k \tag{1a}$$

$$x_a = \sum_k x_{a,k} \qquad \forall\ a \tag{1b}$$

$$x_{a,k} \geqq 0 \qquad \forall\ a,k \tag{1c}$$

[Unless denoted otherwise, "$\sum$" and "$\forall$" include all elements of the relevant set. For instance, in eq. (1b), "$\sum_k$" means "$\sum_{k \in \underset{\sim}{K}}$" and "$\forall$ a" means "$\forall$ $a \in \underset{\sim}{A}$".]

The uncapacitated, multicommodity fixed charge network design problem (denoted FCND) can now be formulated as the following integer program:

Program FCND:

$$\min_{\underline{x} \in \underset{\sim}{X}} \sum_a f_a \cdot y_a + \sum_k \sum_a c_{a,k} \cdot x_{a,k} \tag{2a}$$

Subject to:

$$x_a \leqq u_a^{max} \cdot y_a \qquad \forall\ a \tag{2b}$$

$$y_a \in \{0,1\} \qquad \forall\ a \tag{2c}$$

Let $\underline{y}^* = (\ldots, y_a^*, \ldots)$ denote the optimal arc selection vector, let $\underline{x}^* = (\ldots, x_a^*, \ldots)$ denote the optimal (aggregate) arc flow vector, and let z*[FCND] denote the optimal objective function value of program FCND. [Throughout this paper, the optimal objective function value for any problem P is denoted as $z^*[P]$.]

Objective function (2a) minimizes the total system costs, including both fixed and variable costs, for all arcs and commodities in the network. Carrying the minimization over set $\underset{\sim}{X}$ guarantees that the arc flows for each commodity are restricted to feasible paths in the network. The aggregate "forcing" constraints (2b) ensure that an arc carries flow only if it is selected to be in the network. Constraints (2c) ensure the integrality of the arc selection decision variables, $\{y_a\}$.

Fixed charge network design problems of the type considered here have been widely studied in the literature. Wong (1978) noted that this problem is a generalization of the Steiner tree problem on a network (Dreyfus and Wagner (1972)) which is itself NP-hard (Karp (1975)); thus, the fixed charge network design problem is also NP-hard. This means that (probably) there is no efficient exact procedure for large scale problems. Some of the largest problems for which optimal solutions have been obtained are those reported by Barr, Glover, and Klingman (1981) and Magnanti, Wong, and Mireault (1984). Barr et al. tailored the branch and bound penalty scheme proposed by Driebeek (1966) to obtain optimal solutions to uncapacitated bipartite networks containing 50 source nodes, 150 sinks nodes, and 600 fixed charge arcs. Magnanti et al. used Benders decomposition to obtain optimal solutions for uncapacitated general networks containing 30 nodes and 90 fixed charge arcs. Other exact procedures have been proposed by Gray (1971), Kennington and Unger (1976) and Fisk and McKeown (1981). Malek-Zavarei and Frisch (1972) demonstrated the equivalence between general and bipartite fixed charge network design problems.

For large scale problems, analysts have typically relied on heuristic procedures coupled with lower bound methods. Fixed charge network design heuristics have been developed by Balinski (1961), Kuhn and Baumol (1962), Billheimer and Gray (1973), Los and Lardinois (1982), and Powell and Sheffi (1983). The near-optimality of a heuristic solution can be gaged by comparing its cost with a lower bound to the optimal objective function value. A number of researchers have developed lower bounds that are specifically tailored to fixed charge network design problems. Billheimer and Gray (1973) and Los and Lardinois (1982) based their lower bounds on the marginal penalty of arc exclusion; Geoffrion (1977), Rardin (1982), and Balakrishnan (1984) relied on Lagrangian relaxation techniques; Rardin and Choe (1979), Balakrishnan (1984), and Balakrishnan, Magnanti and Wong (1987) used dual ascent algorithms (i.e., approximate solutions to the dual problem); and Wong (1978), Magnanti and Wong (1981), and (as previously cited) Magnanti, Wong, and Mireault (1984) applied Benders decomposition procedures to develop lower (and upper) bounds.

The lower bound presented in this paper is based on the LP relaxation of program FCND. This relaxation is presented next.

- **Linear Programming Relaxation**

The following paragraphs show that the LP relaxation of program FCND is equivalent to a shortest path program. This means that the LP relaxation is very easy to solve.

The LP relaxation of program FCND is formed by replacing the integrality conditions (2c) with the following nonnegativity constraints:

$$y_a \geqq 0 \qquad \forall\ a \tag{3}$$

Observe the this relaxation is equivalent to the following shortest path program (with flow assignment):

Program SP:

$$\min_{\underline{x}\in\underset{\sim}{X}} \sum_k \sum_a \left(\frac{f_a}{u_a^{max}} + c_{a,k}\right) \cdot x_{a,k} \tag{4}$$

Let $\bar{\underline{x}}^* = (\ldots,\bar{x}_a^*,\ldots)$ denote the optimal (aggregate) arc flow vector and let z*[SP] denote the optimal objective function value of program SP.

The equivalence between program SP and the original LP relaxation (using eq. (1), (2a), (2b), and (3)) can be shown by using Balinski's (1961) observation that constraints (2b) will always be satisfied with equality in the LP relaxation of program FCND. Thus, since $u_a^{max} > 0 \ \forall\ a$, constraints (2b) can be solved explicitly for $\{y_a\}$. Substituting $\{x_a/u_a^{max}\}$ for $\{y_a\}$ yields the formulation given in eq. (4). [Note that constraints (3) can be omitted in SP because the nonnegativity of $\{y_a\}$ is ensured by the fact that the arc flows in set $\underset{\sim}{X}$ are required to be nonegative.]

Because program SP is uncapacitated, it separates by commodity into a series of independent subprograms. Let SP_k denote the subprogram for commodity k. For each k, program SP_k is solved by determining the shortest path from O(k) to D(k) using the linearized cost coefficients given in formulation (4) and sending the entire commodity demand, d_k, over that path. [See Deo and Pang (1984) for a recent survey of shortest path algorithms.] The optimal solution to program SP, then, is simply the aggregation of the individual subprograms. The optimal aggregate flow for an arc is the sum of the optimal commodity-specific flows determined in each of the subprograms (see eq. (1b)); and the optimal objective function value of SP is the sum of the optimal objective function values for each of the subprograms; i.e., $z^*[SP] = \sum_k z^*[SP_k]$.

The analysis conducted later in this paper involves an ensemble of shortest path programs in which the maximum flow parameter vector $\underline{u}^{max} = (\ldots, u_a^{max}, \ldots)$ is altered systematically. Thus, let SP($\underline{u}$) denote the shortest path program of the form of eq. (4) in which the vector $\underline{u} = (\ldots, u_a, \ldots)$ is substituted for $\underline{u}^{max}$. For any given vector $\underline{u}$, program SP($\underline{u}$) is solved by the same method discussed after eq. (4). [In this paper, it will be convienent to use both "SP" and "SP($\underline{u}^{max}$)" to refer to the LP relaxation of program FCND.]

The significance of basing a lower bound on program SP is discussed next.

• Aggregate Versus Disaggregate Formulations

As mentioned earlier, the "forcing" constraints (2b) in program FCND are aggregated (i.e., summed) over commodities. Many researchers, however, have chosen to formulate the fixed charge network design problem using disaggregate "forcing" constraints. In the disaggregate formulation, denoted DFCND, constraints (2b) are replaced with

$$x_{a,k} \leq d_k \cdot y_a \qquad \forall\ a,k \tag{5}$$

Observe that programs FCND and DFCND are equivalent: they have the same feasible region, optimal solution(s), and objective function value. The reason that most analysts have preferred formulation DFCND is that the feasible region of its LP relaxation, denoted DLP, is contained within the feasible region of program SP (the LP relaxation of program FCND). Consequently, the following relationships hold (Rardin 1982):

$$z^*[SP] \leq z^*[DLP] \leq z^*[DFCND] = z^*[FCND] \tag{6}$$

In short, the disaggregate formulation yields a tighter LP relaxation than that of the aggregate model. In fact, in many cases the solution to DLP is also optimal (or nearly optimal) in DFCND (and FCND).

A number of authors have stressed the (apparent) inferiority of aggregate formulations. In their location and distribution model, Geoffrion and Graves (1974) found that an aggregate formulation produced much looser Benders cuts than an alternative disaggregate formulation. For certain classes of uncapacitated facility location problems, Cornuejols, Fisher, and Nemhauser (1977) showed that the LP relaxation of the aggregate version of the problem became successively looser as the problem size increased. Aggregate and disaggregate formulations for fixed charge network design problems have been compared by Wong (1978), Rardin and Choe (1979), Rardin (1982), Balakrishnan (1984), Magnanti, Wong, and Mireault (1984), and Balakrishnan, Magnanti and Wong (1987). Additional discussion of aggregate verses disaggregate formulations is contained in Efroymson and Ray (1966), Davis and Ray (1969), Garfinkel, Neebe, and Rao (1974), Erlenkotter (1978), Guignard and Spielberg (1979), and Magnanti and Wong (1981).

In light of the discussion above, it is somewhat surprising that our lower bound is based on a seemingly weaker aggregate formulation. Our reasons are twofold. First, as shown perviously, program SP($\underline{u}$) is very easy to solve; and second, as shown in the next section, the "capacity" parameter vector, $\underline{u}$, can be adjusted so as to produce bounds that are significantly tighter than the LP relaxation of the aggregate program formulation.

2. CAPACITY IMPROVEMENT PROCEDURE

The capacity improvement (CI) procedure presented in this section is a method of obtaining a lower bound to z*[FCND]. Naturally, since SP is the LP relaxation of FCND, z*[SP] is a lower bound to z*[FCND]. However, as mentioned in connection with eq. (6), because FCND is an aggregate formulation, its LP relaxation will, in general, yield a loose lower bound. A tighter lower bound can be obtained by using a capacity parameter vector, $\underline{u}$, that is smaller than $\underline{u}^{max}$. Clearly, then, $z^*[SP(\underline{u})] \geq z^*[SP]$. The "trick" to this approach is to determine conditions on $\underline{u}$ such that $z^*[SP(\underline{u})] \leq z^*[FCND]$. This is the idea behind the CI procedure described in this section.

The presentation here is divided into four parts. The first part proves the validity of the lower bound developed by the CI procedure. The second part shows that this lower bound is easy to compute because it can be determined by solving a set of linearized knapsack problems. The third part demonstrates how the CI procedure can be used in an iterative fashion to obtain a successively tighter lower bound to z*[FCND]. Finally, the last part presents an algorithm summarizing the CI procedure.

• Lower Bound

The following paragraphs develop a method for obtaining a lower bound to z*[FCND] that is at least as tight as z*[SP]. The approach uses a value t, referred to as the "target value". Suppose, for purposes of discussion, that t is known to be an upper bound to z*[FCND]. This information can be used to determine an upper bound (or "capacity") on the flow of a generic arc, say arc b, in the optimal solution to FCND. Let $\bar{w}_b$ be a trial value of such a "capacity" and consider the program $A_b(\bar{w}_b)$ (referred to as the "auxiliary program for arc b") obtained by adding the constraint

$$x_b \geq \bar{w}_b \tag{7}$$

to program SP, the LP relaxation of FCND. The optimal objective function value of $A_b(\bar{w}_b)$ as a function of $\bar{w}_b$ is shown in Figure 1. If $z^*[A_b(\bar{w}_b)]$ is greater than or equal to t (and t, by assumption, is greater than z*[FCND]), then clearly the flow on arc b cannot exceed $\bar{w}_b$ in any optimal solution to FCND. Thus, $\bar{w}_b$ is a valid capacity for the flow on arc b. The "best" capacity parameter (i.e., the tightest valid upper bound on flow) is identified by finding the minimum of u_b^{max} and the smallest value of $\bar{w}_b$ such that $z^*[A_b(\bar{w}_b)]$ is greater than or equal to t. That is, let

$$w_b(t) = \min \{ \bar{w}_b : z^*[A_b(\bar{w}_b)] \geq t \} \tag{8}$$

(see Figure 1) and define $u_b(t)$, referred to as an "improved capacity parameter for arc b", as

$$u_b(t) = \min \{ w_b(t), u_b^{max} \} \tag{9}$$

[Program $LK_b(t)$, presented shortly, finds such a capacity parameter.]

The procedure described in the preceding paragraph can be performed for each arc $b \in \underset{\sim}{A}$ (or any subset of the arcs contained in $\underset{\sim}{A}$). That is, for each arc $b \in \underset{\sim}{A}$, a separate auxiliary program $A_b(\bar{w}_b)$ is created and eq. (8) and (9) are used to determine the improved capacity parameter for that arc. This, then, determines an improved capacity parameter <u>vector</u>, $\underline{u}(t) = (\ldots, u_b(t), \ldots)$.

Next, consider program $SP(\underline{u}(t))$. [Remember that $SP(\underline{u}(t))$ is the shortest path program defined in eq. (4) with $\underline{u}^{max}$ replaced by $\underline{u}(t)$.] Because $t > z^*[FCND]$, this means that $\underline{u}(t)$ is an upper bound to $\underline{x}^*$ (where $\underline{x}^*$ is the optimal aggregate arc flow vector in FCND) and so the optimal solution to FCND is contained in the feasible region of $SP(\underline{u}(t))$. Thus, $z^*[SP(\underline{u}(t))]$ is a valid lower bound to $z^*[FCND]$.

The discussion above assumed that t was known to be an upper bound to $z^*[FCND]$. Suppose, now, that t is an arbitrarily selected value. Then either t must itself be a lower bound to $z^*[FCND]$ or (if $t > z^*[FCND]$) then $z^*[SP(\underline{u}(t))]$ must be a valid lower bound to $z^*[FCND]$. This information can be combined to define the "capacity improvement (CI) lower bound", $\ell(t)$. Specifically,

$$\ell(t) = \min \{ t, z^*[SP(\underline{u}(t))] \} \qquad (10)$$

Note, in addition, that since $u_b(t)$ is less than or equal to u_b^{max} for all arcs (see eq. (9)), then $z^*[SP(\underline{u}(t))]$ must be greater than or equal to $z^*[SP]$, the optimal objective function value of the original LP relaxation. Thus, if t is greater than or equal to $z^*[SP]$, then so is $\ell(t)$.

In summary, the above discussion has proved the following proposition (Lamar (1985)):

Proposition 1:

If $t \geqq z^*[SP]$, then $z^*[SP] \leqq \ell(t) \leqq z^*[FCND]$

An efficient method for determining $\ell(t)$ is described next.

- **Knapsack Interpretation**

The following paragraphs show how the improved capacity parameter vector, $\underline{u}(t)$, can be obtained by solving a set of linearized knapsack programs. Since such programs can be solved by a "greedy-type" algorithm, this means that $\underline{u}(t)$ can be determined very efficiently.

This procedure can be described once again by considering a generic arc $b \in \underset{\sim}{A}$. Remember that the intermediate capacity parameter, $w_b(t)$, is obtained by adding constraint (7) to program SP. To describe the effect of this constraint, let $\Delta_{b,k}$ denote the marginal cost difference between the following two quantities:

(i) the optimal LP routing of a unit of commodity k from O(k) to D(k) using arc b; and

(ii) the current LP routing of a unit of commodity k from O(k) to D(k) (which is optimal in program SP without the constraint that arc b be used).

[The calculation of $\Delta_{b,k}$ is described at the end of this subsection.]

Now consider the following linear program which is also focused on a particular arc b.

Program $LK_b(\bar{t})$:

$$\max \sum_k d_k \cdot r_{b,k} \tag{11a}$$

Subject to:

$$\sum_k (d_k \cdot \Delta_{b,k}) \cdot r_{b,k} \leq \bar{t} - z*[SP] \tag{11b}$$

$$0 \leq r_{b,k} \leq 1 \qquad \forall\ k \tag{11c}$$

The decision variables $\{r_{b,k}\}$ give the proportion of commodity k that is routed via arc b; and $\bar{t}$ is a parameter of the program. Figure 2 depicts the optimal objective function value of $LK_b(\bar{t})$ as a function of $\bar{t}$. Observe that this program is a linearized 0-1 knapsack program and thus can be solved simply be ranking the $|\underset{\sim}{K}|$ marginal costs $\{\Delta_{b,k}\}$ in increasing order (Dantzig (1957)). By using efficient sorting techniques (see, for example, Ahrens and Finke (1975)), this program can be solved very fast.

The purpose of program $LK_b(\bar{t})$ is to determine the value of $w_b(t)$. Specifically,

$$w_b(t) = z*[LK_b(t)] \tag{12}$$

To see this relationship, compare Figures 1 and 2. As mentioned earlier, Figure 1 shows the parametric analysis of $z*[A_b(\bar{w}_b)]$ as $\bar{w}_b$ increases from zero whereas Figure 2 shows the parametric analysis of $z^*[LK_b(\bar{t})]$ as $\bar{t}$ increases from $z^*[SP]$. Note, however, that Figure 2 is simply the "inversion" of the graph in Figure 1. Thus, for any specific target value, t, the optimal objective function value of program $LK_b(t)$ corresponds to the right-hand-side parameter of constraint (7) in program $A_b(w_b(t))$.

Once $w_b(t)$ has been determined using eq. (12), the improved capacity parameter, $u_b(t)$, can be obtained by the minimization given in eq. (9). Similarly, the improved capacity parameter vector, $\underline{u}(t)=(\ldots,u_b(t),\ldots)$, can be computed by solving a separate linearized knapsack program for each arc $b \in \underset{\sim}{A}$. The lower bound, $\ell(t)$, can then be determined using eq. (10).

The determination of the marginal costs $\{\Delta_{b,k}\}$ in program $LK_b(t)$ is now described. For each arc $b \in \underset{\sim}{A}$ and commodity $k \in \underset{\sim}{K}$, the cost of routing, via arc b, one unit of commodity k can be decomposed into the following four components (see Figure 3):

- the linearized unit cost on arc b (13a)
- the shortest path from O(k) to I(b) (13b)
- the shortest path from J(b) to D(k) (13c)
- the shortest path from O(k) to D(k) (13d)

The marginal cost $\Delta_{b,k}$ is then computed as (13a) plus (13b) plus (13c) minus (13d).

Observe that $\Delta_{b,k}$ can be determined directly from the solution of program SP. To see this, let $\{v_{n,k}\}$ denote the optimal dual variables associated with constraints (1a) in program SP. Using these values, the four cost components given above can be reexpressed as follows:

- $(f_b/u_b^{max}) + c_{b,k}$ (13a′)
- $v_{I(b),k} - v_{O(k),k}$ (13b′)
- $v_{D(k),k} - v_{J(b),k}$ (13c′)
- $v_{D(k),k} - v_{O(k),k}$ (13d′)

Combining (13a′) plus (13b′) plus (13c′) minus (13d′) yields the following relationship:

$$\Delta_{b,k} = \frac{f_b}{u_b^{max}} + c_{b,k} + v_{I(b),k} - v_{J(b),k} \qquad (14)$$

Notice, however, that the right-hand-side of eq. (14) is simply the reduced cost for arc b and commodity k and so is directly available from the optimal solution to program SP.

An important feature of the procedure outlined in the preceding paragraphs is that $\ell(t)$ is relatively easy to compute. As described above, the calculation of the coefficients $\{\Delta_{b,k}\}$ is straightforward and the solution of program $LK_b(t)$ can be determined by a greedy-type algorithm. This means that it is easy to determine the vector $\underline{u}(t)$ used in program $SP(\underline{u}(t))$. And, since $SP(\underline{u}(t))$ is itself easy to solve (it is a shortest path program), the lower bound, $\ell(t)$, can be determined very efficiently.

The techniques introduced above can also be used iteratively to generate a successively tighter lower bound. This procedure is discussed next.

• **Iterative Procedure**

The CI procedure can be used within an iterative framework to obtain a successively tighter lower bound to z*[FCND]. The concept here is to use the improved capacity parameter vector from the previous iteration to determine the vector for the current iteration. [Thus, the material presented earlier in this section can be viewed as the initial iteration of this procedure.] Changes in the target value between iterations are allowed, although the procedure requires that the sequence of target values be nonincreasing. [The reason for this restriction is explained at the end of this subsection.]

To describe the iterative process, let i be the iteration counter for i = 1,2,... . Iterations i-1 and i are referred to, respectively, as the "previous" and "current" iterations. Let t^i denote the target value for the current iteration. Because the procedure depends, in general, on the entire sequence of target values (not just the current value), it is useful to define $\underset{\sim}{T}^i = \{t^0,\ldots,t^i\}$ as the current target value <u>set</u>. [Note that the number of elements in this set increases by one in each iteration. Also, by definition, let $t^0 \equiv \infty$; i.e., $\underset{\sim}{T}^0 \equiv \{\infty\}$.] In addition, let $\underline{u}(\underset{\sim}{T}^i)$ denote the current improved capacity parameter vector and let $\ell(\underset{\sim}{T}^i)$ denote the current CI lower bound to z*[FCND]. [By definition, let $\underline{u}(\underset{\sim}{T}^0) \equiv \underline{u}^{max}$ and let $\ell(\underset{\sim}{T}^0) \equiv$ z*[SP].]

Now consider a generic arc $b \in \underset{\sim}{A}$ and let $\Delta^i_{b,k}$ denote the marginal cost of sending, via arc b, a unit of commodity k based on the linearized arc cost coefficients used in program

$SP(\underline{u}(\underset{\sim}{T}^{i-1}))$. Next, consider the following linearized 0-1 knapsack program:

Program $LK_b(\underset{\sim}{T}^i)$:

$$\max \sum_k d_k \cdot r^i_{b,k} \tag{15a}$$

Subject to:

$$\sum_k (d_k \cdot \Delta^i_{b,k}) \cdot r^i_{b,k} \leq t^i - z^*[SP(\underline{u}(\underset{\sim}{T}^{i-1}))] \tag{15b}$$

$$0 \leq r^i_{b,k} \leq 1 \qquad \forall\ k \tag{15c}$$

[Note that, for i=1, the program given above is the same as program $LK_b(t)$ (given in eq. (11)).] The current intermediate capacity parameter for arc b is defined as

$$w_b(\underset{\sim}{T}^i) = z^*[LK_b(\underset{\sim}{T}^i)] \tag{16}$$

and the current improved capacity parameter for arc b is

$$u_b(\underset{\sim}{T}^i) = \min \{\ w_b(\underset{\sim}{T}^i),\ u_b(\underset{\sim}{T}^{i-1})\ \} \tag{17}$$

The current improved capacity parameter <u>vector</u>, $\underline{u}(\underset{\sim}{T}^i) = (\ldots, u_b(\underset{\sim}{T}^i), \ldots)$, is obtained by determining eq. (16) and (17) for each arc $b \in \underset{\sim}{A}$.

Then, the current lower bound, $\ell(\underset{\sim}{T}^i)$, is defined as follows:

$$\ell(\underset{\sim}{T}^i) = \max \left[\ \min \{\ t^i,\ z^*[SP(\underline{u}(\underset{\sim}{T}^i))]\ \},\ \ell(\underset{\sim}{T}^{i-1})\ \right] \tag{18}$$

The two terms in the minimization within the brackets above are analogous to those used in eq. (10); the two terms in the maximization above ensure that the sequence $\{\ell(\underset{\sim}{T}^i)\}$ is nondecreasing.

As mentioned at the beginning of this subsection, the sequence of target values is required to be nonincreasing. The reason for this restriction can be explained by considering the possible consequences of an increasing subsequence of target values. Specifically, suppose that for some i, $t^{i-1} < z^*[\text{FCND}]$ and $t^i > z^*[\text{FCND}]$. Because $t^{i-1} < z^*[\text{FCND}]$, it is possible that $u_b(\underset{\sim}{T}^{i-1}) < x_b^*$ (where x_b^* is the optimal flow on arc b in FCND) for some $b \in \underset{\sim}{A}$ thus making it possible that $z^*[SP(\underline{u}(\underset{\sim}{T}^{i-1}))] > z^*[\text{FCND}]$. Furthermore, because $u_b(\underset{\sim}{T}^i) \leq u_b(\underset{\sim}{T}^{i-1})$ (see eq. (17)) it is also possible that $z^*[SP(\underline{u}(\underset{\sim}{T}^i))] > z^*[\text{FCND}]$. But, if both t^i and $z^*[SP(\underline{u}(\underset{\sim}{T}^i))]$ are greater than $z^*[\text{FCND}]$, then $\ell(\underset{\sim}{T}^i)$ will not be a valid lower bound to $z^*[\text{FCND}]$.

The situation described in the preceding paragraph can be avoided by requiring that the sequence $\{t^i\}$ be nonincreasing. [Note that this is a sufficient, but not necessary, condition to ensure that $\ell(\underset{\sim}{T}^i)$ is a valid lower bound to $z^*[\text{FCND}]$.] Hence, the following proposition:

<u>Proposition 2</u>:

For $i = 1,2,\ldots$, if $t^i \leq t^{i-1}$,

then $z^*[SP] \leq \ell(\underset{\sim}{T}^{i-1}) \leq \ell(\underset{\sim}{T}^i) \leq z^*[\text{FCND}]$

[See Lamar (1985) for additional discussion.] Proposition 2 states that if the sequence of target values $\{t^i\}$ is nonincreasing in the iterative procedure, then the sequence $\{\ell(\underset{\sim}{T}^i)\}$ produced by this procedure contains successively tighter (or at least nondecreasing) lower bounds to z*[FCND]; and that each of these lower bounds is at least as tight as z*[SP], the optimal objective function value of the LP relaxation of FCND. [Section 3 gives several numerical examples which illustrate how the choice of a target value affects the lower bound to z*[FCND] produced by this iterative procedure.]

Another consequence of Proposition 2 is that if for some iteration, say the j^{th}, $\ell(\underset{\sim}{T}^j)$ equals t^j, then no further improvement in the CI lower bound can be obtained. This is because $\{t^i\}$ is nonincreasing and so for all iterations subsequent to the j^{th}, the minimization term in eq. (18) can be no greater then t^j. Thus, the largest value that $\ell(\underset{\sim}{T}^i)$ can obtain is t^j.

The iterative procedure given here forms the basis of the algorithm that is outlined next.

- **Algorithm**

The algorithm described below summarizes the CI procedure presented in this section. It produces a lower bound to z*[FCND] that is at least as tight as the LP relaxation of FCND. The steps of the algorithm are shown in Figure 4. The following paragraphs comment on each of these steps.

Step 0 initializes the algorithm. Here, the shortest path program, SP (see eq. (4)), is solved and the following assignments are made: $i \leftarrow 0$; $\underset{\sim}{T}^0 \leftarrow \{\infty\}$; $\underline{u}(\underset{\sim}{T}^0) \leftarrow \underline{u}^{max}$; and $\ell(\underset{\sim}{T}^0) \leftarrow$ z*[SP].

Step 1 increments the iteration counter.

Step 2 selects t^i, the target value for the current iteration. The only condition imposed on this choice is that $t^i \leq t^{i-1}$. [The "art" of choosing the target value is explored in the examples in the next section.]

Step 3 uses eq. (16) to compute $w_b(\underset{\sim}{T}^i)$ for each arc b. A "greedy-type" algorithm is used to solve each of the linearized knapsack programs, $LK_b(\underset{\sim}{T}^i)$ (see eq. (15)).

Step 4 uses eq. (17) to compute each of the elements (i.e., arcs) in the current improved capacity parameter vector, $\underline{u}(\underset{\sim}{T}^i)$.

Step 5 solves program $SP(\underline{u}(\underset{\sim}{T}^i))$ using a shortest path algorithm. [Note that the optimal path in the previous iteration can be used as the initial path in the current iteration. In many cases, this initial path is also optimal in $SP(\underline{u}(\underset{\sim}{T}^i))$.]

Step 6 determines $\ell(\underset{\sim}{T}^i)$, the current CI lower bound, using eq. (18).

Step 7 tests whether or not to terminate the algorithm. Here, a relative improvement criterion such as

$$\frac{\ell(\underset{\sim}{T}^i) - \ell(\underset{\sim}{T}^{i-1})}{\ell(\underset{\sim}{T}^{i-1})} < \delta \qquad (19)$$

can be used where $\delta > 0$ is a suitably small, prespecified constant. If criterion (19) <u>is</u> satisfied, then the algorithm outputs the current lower bound, $\ell(\underset{\sim}{T}^i)$, and stops. Otherwise, if this criterion is <u>not</u> satisfied, then the algorithm goes to step 1 and performs another iteration.

The next section illustrates the operation of the CI algorithm with several simple examples.

3. NUMERICAL EXAMPLES

This section, using two numerical examples, illustrates the operation of the CI procedure presented in Section 2. The first example shows that the CI lower bound can eliminate the optimality gap associated with the aggregate LP relaxation of the fixed charge network design problem. The second example further illustrates the CI algorithm and suggests a simple method for selecting the sequence of target values. For both of these examples, the optimality gap of the CI lower bound, expressed as a percent, is measured as follows:

$$\left(\frac{z^*[FCND] - \ell(\underset{\sim}{T}^i)}{z^*[FCND]}\right) \cdot 100 \qquad (20)$$

Similarly, expression (20) is used to measure the optimality gap associated with the aggregate and disaggregate LP relaxations by replacing $\ell(\underset{\sim}{T}^i)$ with z*[SP] and z*[DLP], respectively. [Also, note that z*[DFCND] could be substituted for z*[FCND] in (20) because z*[FCND] = z*[DFCND].]

• First Example

The purpose of this example is to demonstrate that the CI lower bound can converge to the optimal objective function value of the aggregate and disaggregate integer programs (i.e., $\ell(\underset{\sim}{T}^i)$ = z*[FCND] = z*[DFCND]). The multicommodity network used here is shown in Figure 5. For convenience, let the number of commodities, $|\underset{\sim}{K}|$, be denoted as $\bar{k}$. For $k=1,2,\ldots,\bar{k}$, commodity k originates at node 0 and terminates at node k. Each demand d_k is assumed to be unity. Arc (0,1) is designated as "arc b". It has a fixed charge of $f_b = 1$ and a routing cost of $c_{b,k} = 1$ for each commodity k. All other arcs have zero fixed charge and zero routing costs. The maximum possible flow on arc b is $\sum_k d_k = \bar{k}$ and so $u_b^{max} = \bar{k}$.

The optimal solution to the fixed charge network design problem for the network in Figure 5 can be obtained by inspection. It consists of sending a unit of flow over each arc (0,k) for $k=1,2,\ldots,\bar{k}$ and zero flow over all other arcs. The optimal objective function value is the cost of sending flow over arc b. Thus, z*[FCND] = z*[DFCND] = z*[DLP] = 2, and $z^*[SP] = 1+(1/\bar{k})$. Note that there is no optimality gap for DLP, but that the gap of SP is $100\cdot(1-(1/\bar{k}))/2$. Observe also that, as $\bar{k}$ increases, so does the optimality gap of SP which, as Cornuejols, Fisher, and Nemhauser (1977) point out, is a weakness of the aggregate LP relaxation for this class of problems.

For any $\bar{k}$, the optimality gap associated with the aggregate LP relaxation can be reduced by using the CI procedure. For purposes of exposition, let $t^i = t \ \forall \ i$ (i.e., $\underset{\sim}{T}^i = \{\infty,t,t,\ldots,t\}$) where t is a constant target value chosen in the range of $1+(1/\bar{k}) < t < 1+\bar{k}$. [Target values outside this range are not meaningful because they produce a CI lower bound that is no better the optimal objective function value of SP.] As mentioned above, only one arc has a fixed charge: arc b. Thus, only one capacity parameter needs to be improved (i.e, reduced): $u_b(\underset{\sim}{T}^i)$. For this simple network, $u_b(\underset{\sim}{T}^i)$ can be expressed as

$$u_b(\underset{\sim}{T}^i) = \frac{t \cdot u_b(\underset{\sim}{T}^{i-1})}{1 + u_b(\underset{\sim}{T}^{i-1})} \tag{21}$$

with, by definition, $u_b(\underset{\sim}{T}^0) = u_b^{max} = \bar{k}$. Eq. (21), a first-order difference equation, can be solved explicitly (see, for example, Strang (1986)). This yields

$$u_b(\underset{\sim}{T}^i) = \frac{t - 1}{1 + \frac{1}{(t)^i} \cdot \left(1 - \frac{t-1}{\bar{k}}\right)} \tag{22}$$

where $(t)^i$ denotes the constant t taken to the i^{th} power.

For this example, $z^*[SP(\underline{u}(\underset{\sim}{T}^i))]$, the optimal objective function value of program $SP(\underline{u}(\underset{\sim}{T}^i))$, equals $(1/u_b(\underset{\sim}{T}^i)) + 1$ and so can be expressed explicitly (after rearranging terms) as

$$z^*[SP(\underline{u}(\underset{\sim}{T}^i))] = \frac{t}{t - 1} - \frac{1}{(t)^i} \cdot \left(\frac{1}{t - 1} - \frac{1}{\bar{k}}\right) \tag{23}$$

Thus, $\ell(\underset{\sim}{T}^i)$, the CI lower bound in the i^{th} iteration, can be expressed as

$$\ell(\underset{\sim}{T}^i) = \min \left\{t, \frac{t}{t - 1} - \frac{1}{(t)^i} \cdot \left(\frac{1}{t - 1} - \frac{1}{\bar{k}}\right)\right\} \tag{24}$$

Figure 6 depicts $\ell(\underset{\sim}{T}^i)$ given in eq. (24) as a function of the constant target value, t, for several iterations of the CI algorithm. This figure illustrates the two commodity case (i.e.,

$\bar{k} = 2$). For $\bar{k} > 2$, a similar set of curves is produced, but the $z^*[SP]$ line is shifted downward. Observe that for any choice of t in the range $1+(1/\bar{k}) < t < 1+\bar{k}$, the CI lower bound is strictly greater than the aggregate LP relaxation.

Eq. (22), (23), and (24) can also easily be evaluated as $i \to \infty$. Specifically, note that for any $\bar{k}$, because $t > 1$, $1/(t)^i \to 0$ as $i \to \infty$. Thus, in the limit, $u_b(\underset{\sim}{T}^\infty) = t - 1$, $z^*[SP(\underline{u}(\underset{\sim}{T}^\infty))] = t/(t - 1)$, and

$$\ell(\underset{\sim}{T}^\infty) = \min \{ t, t/(t - 1) \} \tag{25}$$

Eq. (25) is depicted in Figure 6 as the line with $i = \infty$. This line represents, for any given target value, the maximum value that the CI lower bound can attain. Observe that, in particular, for the "critical" target value of $t = 2$, the CI lower bound converges to the optimal objective function value of the IP program, thus eliminating the optimality gap associated with the aggregate LP relaxation. This is true regardless of the problem size, $\bar{k}$.

The next example further illustrates the CI procedure.

- **Second Example**

This example illustrates how alternative sequences of target values influence the accuracy versus efficiency of the CI procedure. The rules for choosing the target value given below are intended to provide an easy mechanism for generating a nonincreasing sequence of target values (as required by Proposition 2). Naturally, alternative rules could also be used.

Recall that the target value is determined in step 2 of each iteration of the CI algorithm (see Figure 4). The target value for the first iteration, t^1, is obtained by "rounding-up" the solution to program SP (solved in step 0). Specifically,

$$t^1 = \sum_a f_a \cdot \lceil \bar{y}_a^* \rceil + \sum_k \sum_a c_{a,k} \cdot \bar{x}_{a,k}^* \tag{26}$$

where $\bar{x}_{a,k}^*$ is the optimal flow of commodity k on arc a in program SP, $\bar{y}_a^* = (1/u_a^{max}) \cdot \sum_k \bar{x}_{a,k}^*$, and $\lceil \bar{y}_a^* \rceil$ is the "ceiling" function of $\bar{y}_a^*$ (i.e., the smallest integer greater than or equal to $\bar{y}_a^*$). The target value for the i^{th} iteration (i = 2,3,...) is determined by taking a convex combination of the target value and lower bound from the previous iteration. That is,

$$t^i = \theta \cdot t^{i-1} + (1-\theta) \cdot \ell(\underset{\sim}{T}^{i-1}) \tag{27}$$

where θ is a "discounting" factor in the range $0 < \theta < 1$. Eq. (27) guarantees that, for any allowable θ, the sequence $\{t^i\}$ is nonincreasing. If θ is close to one, then the target values decrease slowly between iterations; if θ is close to zero , then the target values decrease more rapidly. Thus, the CI algorithm can be run several times, each time with a different value of θ, to measure the effect of a slow versus a rapid decrease in the sequence of target values.

The target value selection rules outlined in eq. (26) and (27) were tested on a fixed charge network design problem comprised of $|\underset{\sim}{N}| = 5$ nodes, $|\underset{\sim}{A}| = 20$ fixed charge, uncapacitated arcs, and $|\underset{\sim}{M}| = 20$ commodities (i.e., there was an arc and a commodity "market" for each pair of nodes in the network). [The test network was limited to this size so that the IP and the disaggregate LP could be solved exactly using a commercially available simplex code.] For each arc $a \in \underset{\sim}{A}$, the routing cost $(c_{a,k})$ for all commodities $k \in \underset{\sim}{K}$ and the fixed charge (f_a) was drawn randomly from a uniform distribution, UNIFORM(0,10). Also, for each commodity $k \in \underset{\sim}{K}$, the demand (d_k) was drawn randomly from UNIFORM(0,10). The maximum flow coefficient (u_a^{max}) was taken as the total demand in the network; that is, $u_a^{max} = \sum_k d_k \ \forall \ a$.

Nine "random" networks were generated using the procedure described above. For each of these networks, the IP and the aggregate and disaggregate LP relaxations were solved. In each case, DLP, the disaggregate LP, had a zero percent optimality gap indicating that, for this class of problems, the disaggregate relaxation is exceedingly tight. In contrast, the optimality gap for SP, the aggregate LP, ranged from 7 to 13 percent with a median gap of nearly 11 percent.

For purposes of illustration, the CI procedure using the sequence of target values prescribed by eq. (26) and (27) was applied to the network that produced the median optimality gap for the aggregate LP. The results of this analysis are shown in Figure 7. The figure depicts the CI lower bound, $\ell(\underset{\sim}{T}^i)$, versus the iteration counter, i, for several selected values of θ. As reference points, the figure shows that z*[SP] = 389 and z*[FCND] = 436. In each run of the CI procedure, the algorithm started with $\ell(\underset{\sim}{T}^0)$ = z*[SP] and terminated when $\ell(\underset{\sim}{T}^i) = t^i$. The figure shows, on the one hand, when the target values decrease more slowly (i.e., θ is nearer to 1), the lower bound increases more slowly but the algorithm terminates with a tighter lower bound. For instance, for θ = 0.99, the CI algorithm terminates with a lower bound of $\ell(\underset{\sim}{T}^i)$ = 428 (i.e., 1.8 percent from optimal) after i = 45 iterations. On the other hand, when the target values decrease more rapidly, the converse is true. For instance, for θ = 0.80, the algorithm terminates with $\ell(\underset{\sim}{T}^i)$ = 421 (i.e., 3.4 percent from optimal) after only i = 3 iterations. For this example, a discount factor of θ=0.95 seems to acheive a good balance between the rate of convergence and the size of the optimality gap. In general, the choice of the target value provides for considerable flexibility in the operation of the CI algorithm.

The next section concludes the paper.

4. SUMMARY

This paper, using a capacity improvement (CI) procedure, developed a lower bound for fixed charge, multicommodity, uncapacitated network design problems. Perhaps the most distinctive feature of this procedure is that it is based on an aggregate—rather than disaggregate—problem formulation. In other words, the "forcing" constraints (which enforce logical relationships between the fixed charge-related and the flow-related decision variables) are combined to the greatest extent possible. Thus, compared to its disaggregate counterpart, the aggregate form of the integer program has fewer constraints, but its LP relaxation generally produces a weaker (i.e., looser) lower bound. The purpose of the CI procedure is to tighten this weaker bound.

An algorithm for the CI procedure was presented in this paper. This algorithm, comprised of shortest path and linearized knapsack programs, can be used iteratively to obtain a successively tighter lower bound.

Theoretical and numerical results for the CI lower bound were also presented. The theoretical material showed that the CI procedure produces a lower bound to the fixed charge network design problem that is at least as tight as the LP relaxation of the aggregate integer program. The numerical results demonstrated that CI lower bound can (i) be strictly tighter than the aggregate LP relaxation, (ii) converge to the optimal objective function value of the integer program, and (iii) be adjusted by the sequence of "target values" to tradeoff accuracy versus computational effort.

Besides network design, the CI procedure can also be used to obtain bounds to other fixed charge problems as well. To apply the CI procedure, though, requires an efficient implementation scheme, such as the set of linearized knapsack programs used to determine the CI lower bound for program FCND. Moreover, the efficiency of the CI procedure must be compared to that of techniques based on alternative disaggregate formulations of the problem. Since the LP relaxation of a disaggregate formulation is, in many cases, very tight (see discussion at end of Section 1), methods such as dual ascent and Lagrangian relaxation are frequently very efficient ways of (approximately) solving disaggregate relaxations. Thus, the aggregate-based CI procedure presented in this paper is intended simply as one additional "tool" available to reseachers for developing bounds to certain classes of integer programming problems.

ACKNOWLEDGMENTS

This work was supported, in part, by a research contract from I.U. International to the Massachusetts Institute of Technology and Princeton University. Additional computational funds were provided by the Institute of Transportation Studies and the Graduate School of Management, University of California, Irvine. The authors would like to acknowledge the many thoughtful suggestions—including the knapsack interpretation of the CI procedure—given by the referees of this article. Their comments have resulted in an improved paper.

REFERENCES

J.H. Ahrens and G. Finke (1975), "Merging and Sorting Applied to the 0-1 Knapsack Problem", Operations Research, vol. 23, pp. 19-32.

A. Balakrishnan (1984), "Valid Inequalities and Algorithms for the Network Design Problem with an Application to LTL Consolidation", unpublished Ph.D. dissertation, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

A. Balakrishnan, T.L. Magnanti, and R.T. Wong (1987), forthcoming.

M.L. Balinski (1961), "Fixed-Cost Transportation Problems", Naval Research Logistics Quarterly, vol. 8, pp. 41-54.

R.S. Barr, F. Glover, and D. Klingman (1981), "A New Optimization Method for Fixed Charge Transportation Problems", Operations Research, vol. 29, pp. 448-463.

J.W. Billheimer and P. Gray (1973), "Network Design with Fixed and Variable Cost Elements", Transportation Science, vol. 7, pp. 49-74.

G. Cornuejols, M. Fisher, and G.L. Nemhauser (1977), "Location of Bank Accounts to Optimize Float: An Analytical Study of Exact and Approximate Algorithms", Management Science, vol. 23, pp. 789-810.

G.B. Dantzig (1957), "Discrete-Variable Extremum Problems", Operations Research, vol. 5, pp. 266-276.

P.S. Davis and T.L. Ray (1969), "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem", Naval Research Logistics Quarterly, vol. 16, pp. 331-344.

N. Deo and C.Y. Pang (1984), "Shortest-Path Algorithms: Taxonomy and Annotation", Networks, vol. 14, pp. 275-323.

S.E. Dreyfus and R.A. Wagner (1972), "The Steiner Problem in Graphs", Networks, vol. 1, pp. 195-207.

N.J. Driebeek (1966), "An Algorithm for the Solution of Mixed Integer Programming Problems", Management Science, vol. 12, pp. 576-587.

M.A. Efroymson and T.L. Ray (1966), "A Branch-Bound Algorithm for Plant Location", Operations Research, vol. 14, pp. 361-368.

D. Erlenkotter (1978), "A Dual-Based Procedure for Uncapacitated Facility Location", Operations Research, vol. 26, pp. 992-1009.

J. Fisk and P. McKeown (1981), "The Pure Fixed Charge Transportation Problem", Naval Research Logisitics Quarterly, vol. 26, pp. 631-642.

R.S. Garfinkel, A.W. Neebe, and M.R. Rao (1974), "An Algorithm for the M-Median Plant Location Problem", Transportation Science, vol. 24, pp. 1622-1630.

A.M. Geoffrion (1977), "How Can Specialized Discrete and Convex Optimization Methods be Married?", in Annals of Discrete Mathematics 1: Studies in Integer Programming, P.L. Hammer, E.L. Johnson, B.H. Korte, G.L. Nemhauser (eds.), Elsevier/North-Holland Publ. Co., New York/Amsterdam, Holland, pp. 205-220.

A.M. Geoffrion and G.W. Graves (1974), "Multicommodity Distribution System Design by Benders Decomposition", Management Science, vol. 20, pp. 822-844.

P. Gray (1971), "Exact Solution of the Fixed-Charge Transportation Problem", Operations Research, vol. 19, pp. 1529-1538.

M. Guignard and K. Spielberg (1979), "A Direct Dual Method for the Mixed Plant Location Problem with Some Side Constraints", Mathematical Programming, vol. 17, pp. 198-228.

R.M. Karp (1975), "On the Computational Complexity of Combinatorial Problems", Networks, vol. 5, pp. 45-68.

J. Kennington and E. Unger (1976), "A New Branch-and-Bound Algorithm for the Fixed-Charge Transportation Problem", Management Science, vol. 22, pp. 1116-1126.

H.W. Kuhn and W.J. Baumol (1962), "An Approximate Algorithm for the Fixed-Charges Transportation Problem", Naval Research Logistics Quarterly, vol. 9, pp. 1-15.

B.W. Lamar (1985), "Network Design Algorithms with Applications to Freight Transportation", unpublished Ph. D. dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

M. Los and C. Lardinois (1982), "Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem", Transportation Research-B, vol. 16B, pp. 89-124.

T.L. Magnanti and R.T. Wong (1981), "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria", Operations Research, vol. 29, pp. 464-484.

T.L. Magnanti and R.T. Wong (1984), "Network Design and Transportation Planning: Models and Algorithms", Transportation Science, vol. 18, pp. 1-55.

T.L. Magnanti, R.T. Wong, and P. Mireault (1984), "Tailoring Benders Decomposition for Uncapacitated Network Design", Working Paper No. OR-127-84, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

M. Malek-Zavarei and I.T. Frisch (1972), "On the Fixed Cost Flow Problem", International Journal of Control, vol. 16, pp. 1-23.

W.B. Powell and Y. Sheffi (1983), "The Load Planning Problem of Motor Carriers: Problem Description and Proposed Solution Approach", Transportation Research-A, vol. 17A, pp. 471-480.

R.L. Rardin (1982), "Tight Relaxations of Fixed Charge Network Flow Problems", Report No. J-82-3, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

R.L. Rardin and U. Choe (1979), "Tighter Relaxations of Fixed Charge Network Flow Problems", Report No. J-79-18, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

G. Strang (1986), Introduction to Applied Mathematics, Wellesley-Cambridge Press, Wellesley, MA.

R.T. Wong (1978), "Accelerating Benders Decomposition for Network Design", unpublished Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

LIST OF FIGURES

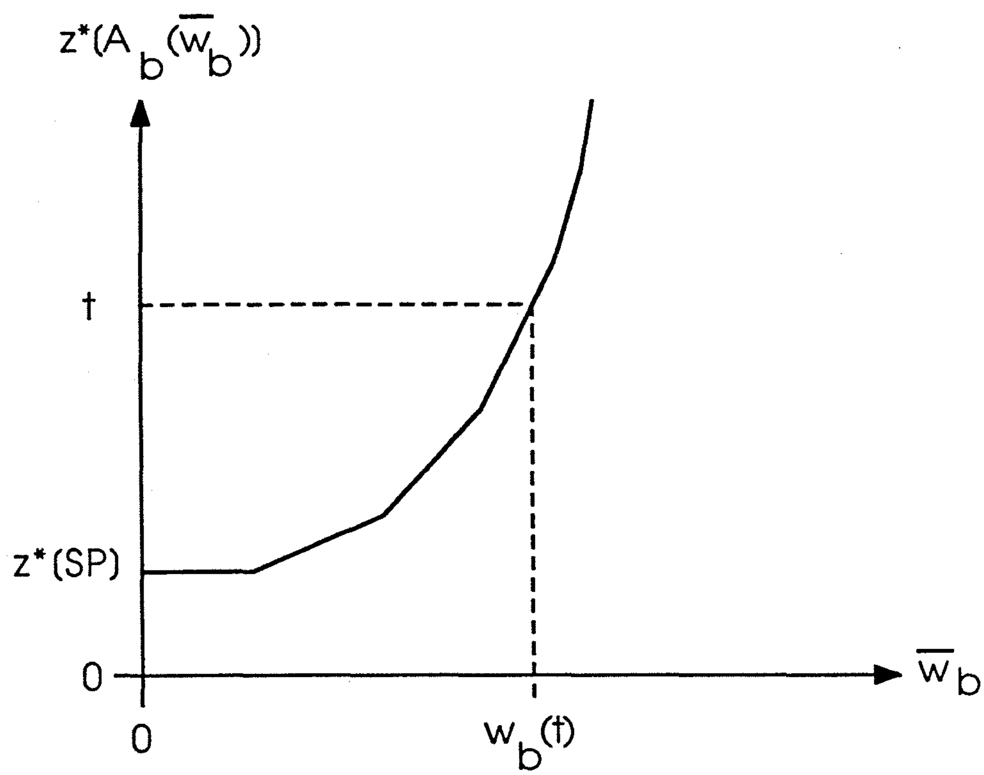

Figure 1

Parametric Analysis of Auxiliary Program

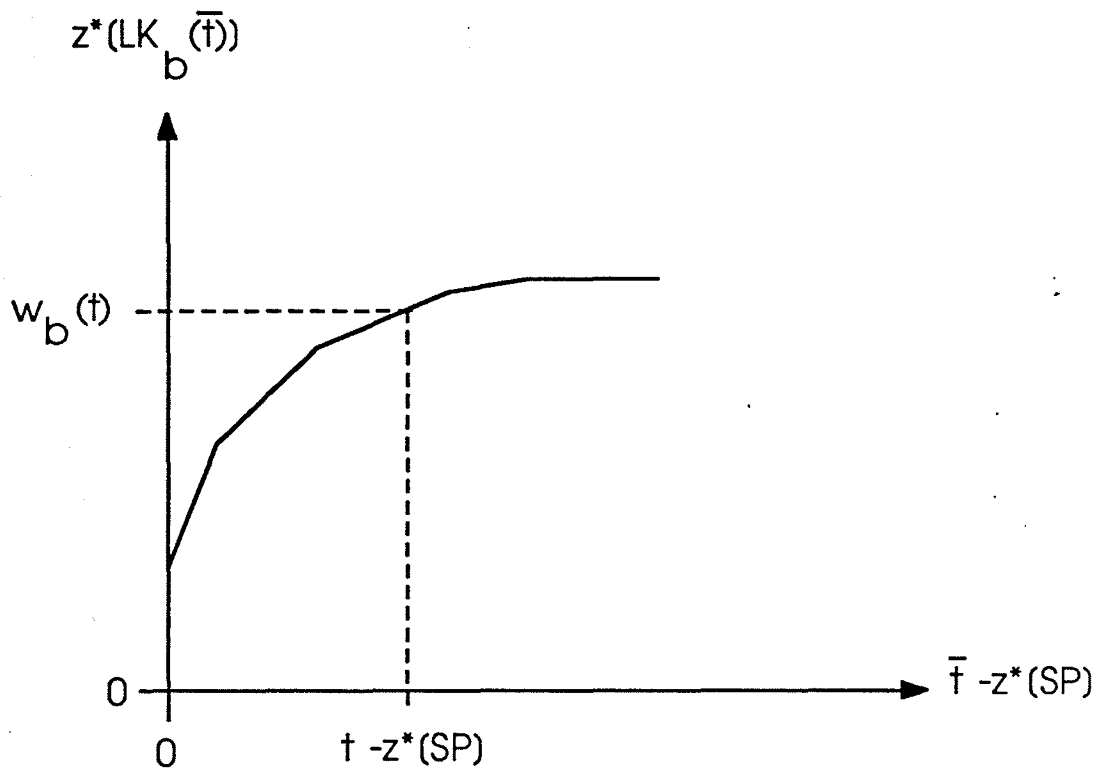

Figure 2

Parametric Analysis of Linearized Knapsack Program

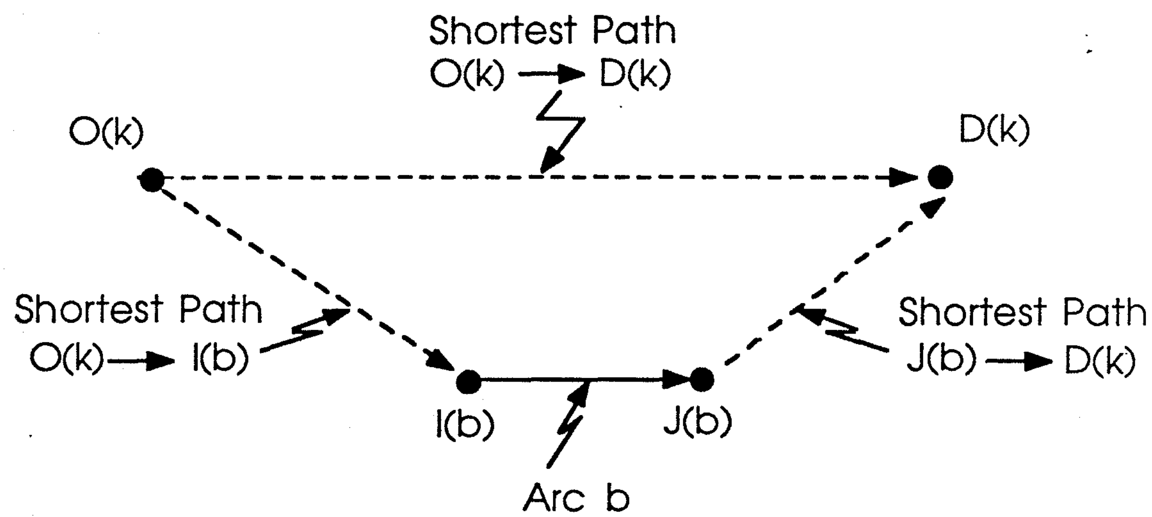

Figure 3

Components of Marginal Cost for Arc b

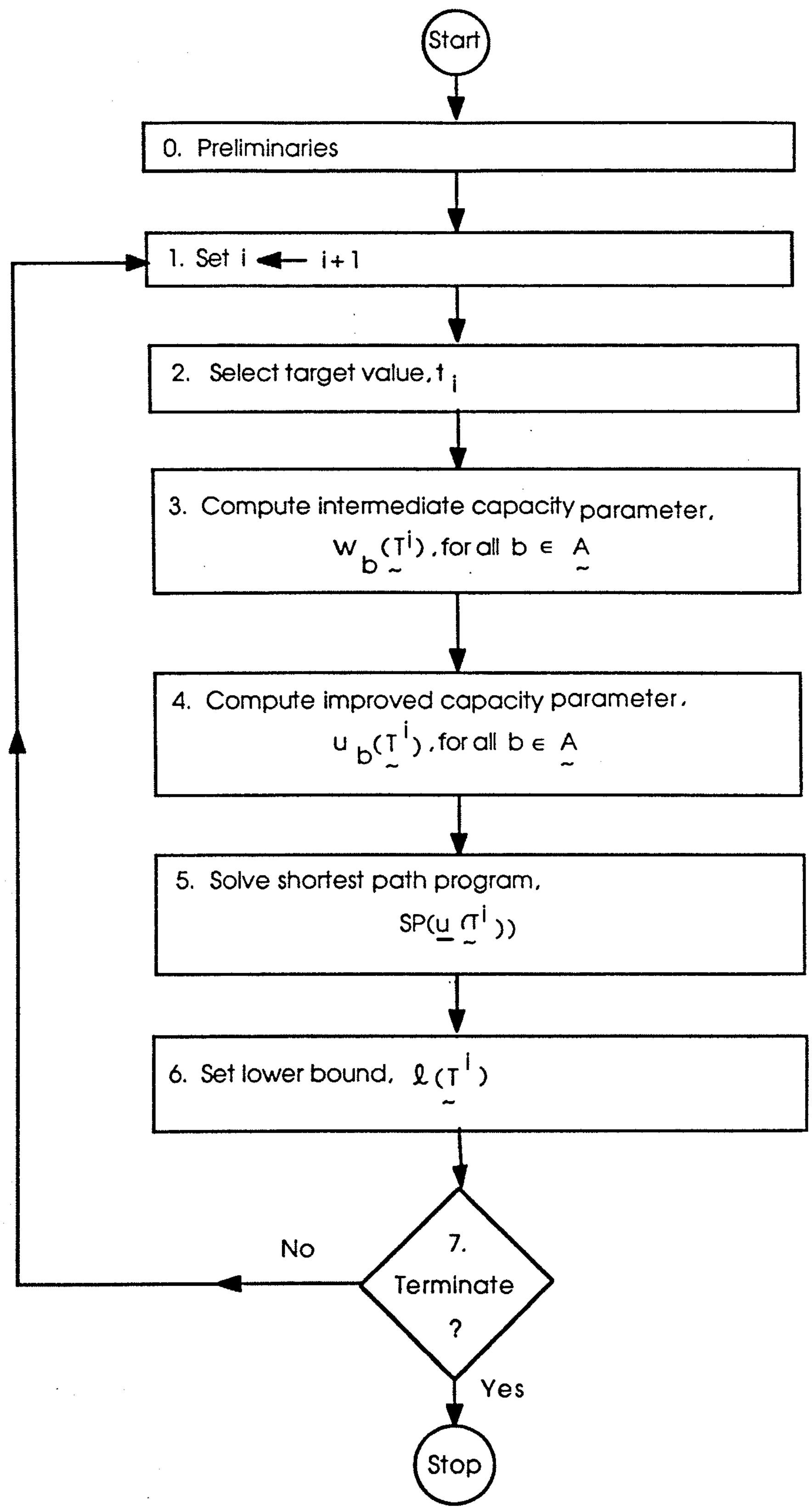

Figure 4

Flowchart of Capacity Improvement Algorithm

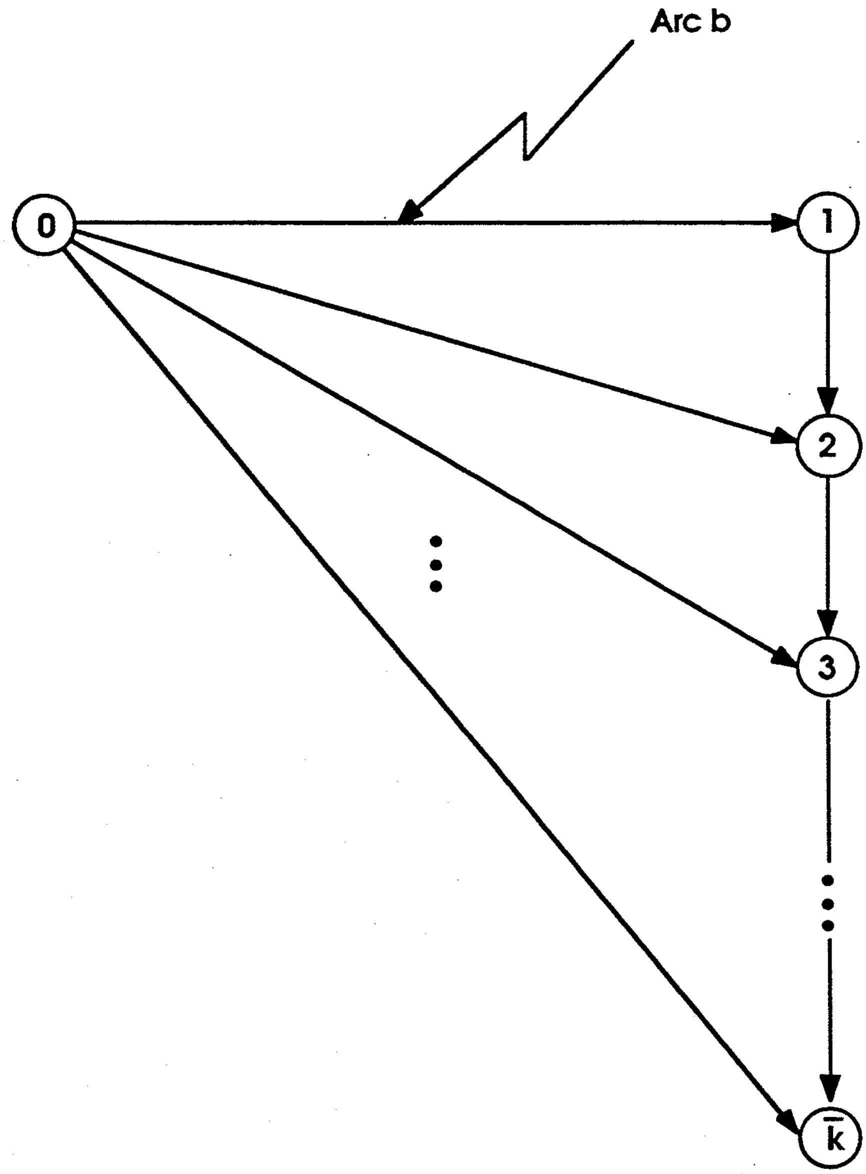

Figure 5

Network for First Example

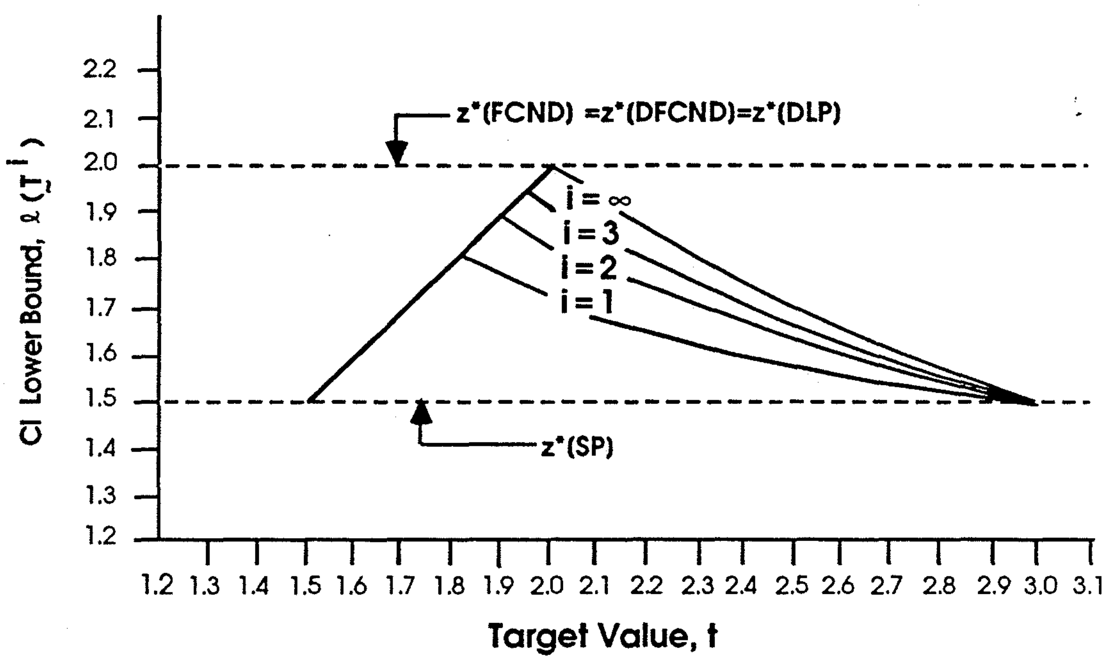

Figure 6

Capacity Improvement Lower Bound for First Example

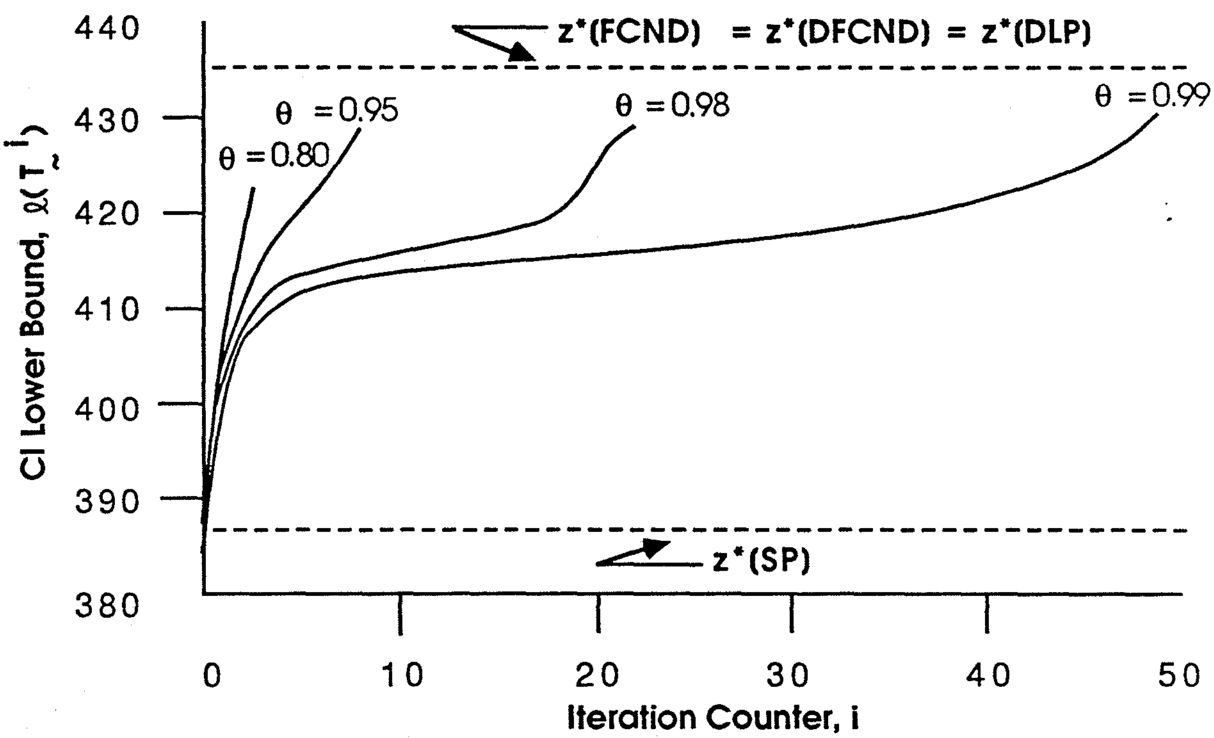

Figure 7

Capacity Improvement Lower Bound for Second Example