A Review of Sensitivity Results for Linear Networks and a New Approximation to Reduce the Effects of Degeneracy

WARREN B. POWELL

Princeton University, Princeton, New Jersey 08544

Estimating the reduced cost of an upper bound in a classical linear transshipment network is traditionally accomplished using the shadow price for this constraint, given by the standard calculation $\bar{c}_{ij} = c_{ij} + \pi_j - \pi_i$. This reduced cost is only a subgradient due to network degeneracy and often exhibits errors of 50% or more compared to the actual change in the objective function if the upper bound were raised by one unit and the network reoptimized. A new approximation is developed, using a simple modification of the original reduced cost calculation, which is shown to be significantly more accurate. This paper summarizes the basic theory behind network sensitivity, much of which is known as folklore in the networks community, to establish the theoretical properties of the new approximation. The essential idea is to use least-cost flow augmenting paths in the basis to estimate certain directional derivatives which are used in the development of the approximation. The technique is motivated with an application to pricing in truckload trucking.

Dynamic networks have proved to be a powerful tool in transportation for optimizing the flows of goods and vehicles over time. The development of efficient network simplex algorithms in the seventies has allowed the development of large models that can be optimized quickly and reliably. These models have for the most part focused on the problem of determining how to route shipments and vehicles over time. Relatively less attention has been devoted to sensitivity issues where the network model is used to estimate the marginal value (or cost) of a unit of flow moving over the network. Specifically, it would be useful to know the marginal value of a unit of freight to the system. For example, we can estimate the total system impact of a load by introducing an additional load to the network model and then reoptimizing completely. If total system costs increase by \$180 after adding the load, then it is necessary to charge a price of at least \$180 to ensure that the load is being carried profitably on a marginal basis.

Section 1 of this paper more thoroughly motivates these sensitivity questions in the context of truckload trucking, but the issues are quite general. The specific problem addressed in this research is the difficulty posed by network degeneracy when using the duals from the network model to solve sensitivity problems.

Consider the classical linear network problem:

$$\text{(P)} \quad \text{Min} \quad F(x) = c^T x \tag{1.1}$$

subject to:

$$Ax = b \tag{1.2}$$

$$x \leqslant u \tag{1.3}$$

$$x \geqslant 0 \tag{1.4}$$

where c is a vector of arc coefficients, x is a vector of link flows, A is a standard node-arc incidence matrix and b is the vector of surpluses and deficits. The vector u represents the set of upper bounds on arcs, where in certain contexts (described in Section 1) an element u_{ij} may represent the forecasted market demand from node i (representing a given region and time period in a dynamic network) to node j. Let π be the vector of duals for the constraints (1.2) and ν the vector of duals for the constraints (1.3), where it is well known that if $x_{ij} = u_{ij}$, then $\nu_{ij} = c_{ij} + \pi_j - \pi_i$ (note that we use the convention for the right hand side of Equation 1.3 that b_i = flow out minus flow in; this is somewhat nonstandard but this will prove later to be very convenient in the context of dynamic networks). If the optimal network bias for (P) is degenerate (as is virtually always the case) then the vector ν is a subgradient of $F(x)$ and only approximates the impact of

Transportation Science
Vol. 23, No. 4, November 1989

0041-1655/89/2304-0231 \$01.25

increasing an element of u. This approximation is sometimes extremely inaccurate and poses practical problems in terms of using these numbers in the field.

The basic theory behind network degeneracy and its effect on the duals and even how to circumvent problems of degeneracy are well understood by the networks community. For the most part, however, certain important properties of networks are not presented in the research literature in a sensitivity context, and some "known" results do not appear to have been reported formally in the literature. For this reason, a summary of major results are given in Section 2, including an important characterization theorem for sensitivity questions for networks. Section 2 also establishes the following results:

i) ν is a subgradient of $F(x^*)$ with respect to the upper bounds u (Proposition 1).
ii) π is a subgradient of $F(x^*)$ with respect to the right hand side b (Corollary 1).
iii) Let D_{ij}^+ be the exact change in $F(x^*)$ if an upper bound u_{ij} is increased by 1. Then
   - $\nu = D_{ij}^+$ if and only if the optimal network basis is not degenerate;
   - if one basis is degenerate, then all alternative optimal bases are degenerate;
   - there does not exist a single vector π which satisfies $\nu_{ij} = D_{ij}^+$ for all links (i, j).
iv) The impact of increasing a right hand side element b_i is a least cost flow augmenting path from i to the root node.
v) the impact of raising an upper bound u_{ij} is a least cost flow augmenting path from j back to i.

These results are used to motivate the following simple approximation. Instead of using $\nu_{ij} = c_{ij} + \pi_j - \pi_i$ to approximate D_{ij}^+, Section 3 proposes using an approximation $\hat{D}_{ij}^+ = c_{ij} + \pi_j^+ - \pi_i^-$, where π^+ and π^- measure the value of one more or one less unit of flow at node i. Section 4 shows that this approximation is easy to calculate and can be extremely accurate.

1. MOTIVATION

PRICING AND load evaluation for truckload motor carriers is the problem of estimating the marginal profit derived from moving an additional unit of freight (see POWELL et al.[16] for a more complete discussion of the issues surrounding pricing and load evaluation). The problem can be viewed as finding a gradient to an appropriate objective function, but the objective function normally used, a linear network transshipment problem, is piecewise linear and therefore not differentiable, creating problems with finding the correct subgradient that answers the question. In addition, there are practical constraints on the speed with which these calculations must be performed that prevent even the use of sophisticated reoptimization schemes. Our problem is to develop an extremely efficient method for quickly and accurately estimating the marginal value of moving an additional unit of freight.

The problem faced by truckload motor carriers is referred to in the literature as the dynamic vehicle allocation (DVA) problem, where carriers must manage a large fleet of vehicles over time to maximize total expected profits. The basic operation in a truckload context works as follows. A shipper will call in a load to be carried, for example, from Pittsburgh on Tuesday arriving in Dallas on Thursday at a specified price. If the carrier accepts the load, it must send a driver to the shipper to pick up the load. After pulling the load from Pittsburgh to Dallas, the truck is emptied and then the carrier must manage the empty vehicle. When a vehicle is empty, the carrier must at any point in time choose whether to assign the vehicle to a new load, move the vehicle empty to a more productive region, or simply hold the vehicle in its current location in anticipation of loads that will arrive later in the day or tomorrow.

The question is better understood by first reviewing the basic optimization problem faced by truckload motor carriers. A relatively straightforward approach to this problem is to formulate it as a dynamic network model such as that shown in Figure 1, which can be represented mathematically using Equations 1.1–1.4. The network is characterized by loaded movement links, which carry a negative arc cost representing the

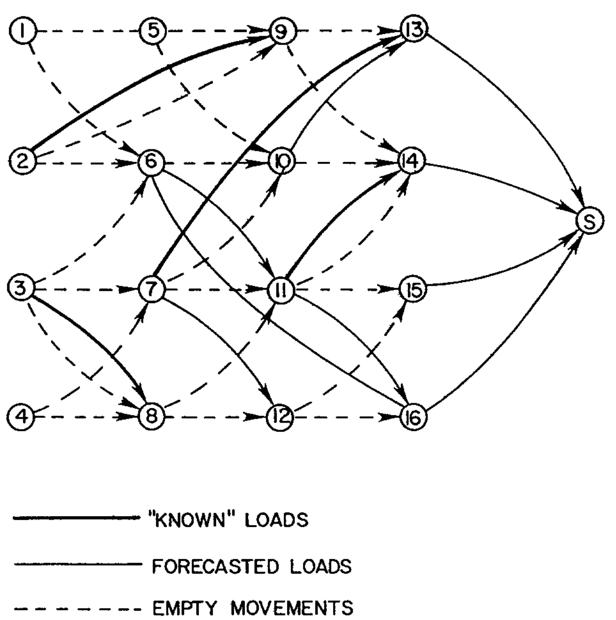

Fig. 1. Illustration of a dynamic network for the vehicle allocation problem.

contribution or profit generated by the load and have an upper bound equal to the market demand between two points. In addition, empty movement links, with a positive arc cost and no upper bound, allow the model to reposition trucks empty in order to take loads in the future. Finally, it is necessary to truncate the planning horizon with a set of "sink links" into a final supersink node. These links may or may not carry an upper bound, and carry a cost that approximates the "salvage value" of a truck terminating in a particular region.

The dynamic vehicle allocation problem has been studied by a number of authors. This problem has been formulated as a static transportation problem (ALLMAN,[1] MISRA[10]), but the more interesting formulations explicitly represent the dynamic aspects. This has been modeled as a deterministic transshipment network (WHITE,[21] WHITE and BOMBERAULT,[22] SHAN[20]), with several authors introducing stochastic elements (JORDAN and TURNQUIST,[8] POWELL et al.,[13] and POWELL[14,15]). Reviews of fleet management models are given in BOOKBINDER and SETHI[2] and DEJAX and CRAINIC,[4] with a specific review of alternative formulations of the DVA, covering both deterministic and stochastic formulations, given in Powell.[15]

This literature has focused primarily on the problem of managing a fleet of vehicles over time. This paper is more concerned with the separate problem of deciding whether to accept a load and, in the longer range, what price to charge for moving the load. The ability to turn down freight (an option that is exercised with care) derives from the relatively small size of truckload carriers, with the largest carrier having approximately one percent of the total market. Large shippers routinely work with several carriers, implying that carriers are routinely refusing freight due to lack of capacity. Thus, when a shipper calls in with a load, a carrier sometimes has the flexibility to evaluate the economics of the load in real time and, if unprofitable, refuse the freight or try to schedule pickup for a later day (when the economics may change). This evaluation must be performed instantly and often on a highly congested computer. One simple approach that has been applied by at least one carrier is to periodically optimize the network in Figure 1 and to maintain a file of network duals π. When a shipper calls in a load that would move from node i to node j, contributing revenue r_{ij} by costing c_{ij} to move, then an estimate of the marginal impact on profits of this load is given by $\bar{p}_{ij} = -\bar{c}_{ij} = -[(c_{ij} - r_{ij}) + \pi_j - \pi_i]$ (since problem (P) minimizes cost, and we wish to estimate the profitability of the load, we use the negative of the reduced cost $\bar{c}_{ij}$). This calculation can be performed instantly by accessing the file of network duals and performing a trivial calculation. The problem is that, as a result of network degeneracy, estimating reduced costs using the vector of duals π from the optimized network will often be quite inaccurate. Five to ten percent of the reduced costs can be sufficiently inaccurate that people in the field will simply reject the entire process.

An alternative to using the reduced costs is to completely reoptimize the network. However, the software driving the screens used by the people talking to the shippers is typically written in COBOL and is written and maintained by an in-house MIS staff. Interfacing such software with sophisticated optimization algorithms is no small task.

Separate from this real-time problem of load evaluation is the longer range problem of pricing, which requires knowing the profitability of a load even if the load must be accepted (for contractual reasons). A powerful approach to this problem (actively used by several carriers) is to calculate the marginal profit for each load, and develop averages for each traffic lane (region to region pair). The marginal profit for each load from a given region r to a given region s can be calculated as an average of all the marginal profit calculations, $\bar{p}_{ij}$, where node i corresponds to region r and node j corresponds to region s. This lane average provides a useful measure of the profitability of freight moving in this lane, and can be used to determine the lowest price that can be charged to insure that freight remains profitable on the margin. Again, however, degeneracy issues in the calculation of marginal profits can distort these numbers, requiring the use of averages of large numbers of loads to minimize the errors.

2. SUMMARY OF THE THEORY OF NETWORK DEGENERACY

DESPITE THE vast literature on linear network flows, there is surprisingly little attention given explicitly to sensitivity problems for networks. The recent books on networks, for example KENNINGTON and HELGASON,[9] PHILLIPS and GARCIA-DIAZ,[12] JENSEN and BARNES[7] and ROCKAFELLAR[18] are devoted almost entirely to algorithms for solving network problems. Typically sensitivity analysis is limited to standard adaptations of results from duality theory for linear programming.

The purpose of this section is to briefly summarize certain sensitivity results for networks. Assume that we have just solved the pure network problem:

$$\text{Min} \quad F(x) = c^T x \tag{2.1}$$

subject to:

$$Ax = b \tag{2.1a}$$

$$x \leqslant u \tag{2.1b}$$

$$x \geqslant 0. \tag{2.1c}$$

Throughout this discussion we adopt the convention that

$$b_i = \text{flow out of node } i - \text{flow into node } i$$

which is the reverse of the standard network convention. From the perspective of dynamic networks, however, this convention will significantly simplify the discussion that follows.

We are interested primarily in the sensitivity of $F(x)$ with respect to the upper bounds. Let x^* be the optimal solution to (P) and define a perturbed problem:

$$(\mathrm{P}_{ij}(\epsilon)) \quad \text{Min} \quad F_{ij}(x) = c^T x(\epsilon) \tag{2.2}$$

subject to:

$$Ax(\epsilon) = b \tag{2.2a}$$

$$x(\epsilon) \leq u + \epsilon e_{ij} \tag{2.2b}$$

$$x(\epsilon) \geq 0 \tag{2.2c}$$

where e_{ij} is a vector of zeroes with a one in the element corresponding to link (i, j). Let $x^*(\epsilon)$ be the optimal solution to p_{ij} and define the directional derivatives:

$$D_{ij}^{+} = \lim_{\epsilon \downarrow 0} \frac{F_{ij}(x^*(\epsilon)) - F(x^*)}{\epsilon} \tag{2.3}$$

and

$$D_{ij}^{-} = \lim_{\epsilon \uparrow 0} \frac{F_{ij}(x^*(\epsilon)) - F(x^*)}{\epsilon}, \tag{2.4}$$

where e_{ij} is the direction the derivative is taken. A typical plot of $F(x^*)$ with respect to u_{ij} is shown in Figure 2, illustrating the familiar piecewise convexity of the problem. If $u_{ij} = 1$, then $F(x^*)$ is not differentiable with respect to u_{ij} and we must talk in terms of subgradients of $F(x^*)$.

Let π be the dual variables for the flow conservation constraints and ν the duals for the link capacity constraints. If $x_{ij}^* = u_{ij}$, then $\nu_{ij} = c_{ij} + \pi_j - \pi_i$. The following result is well known:

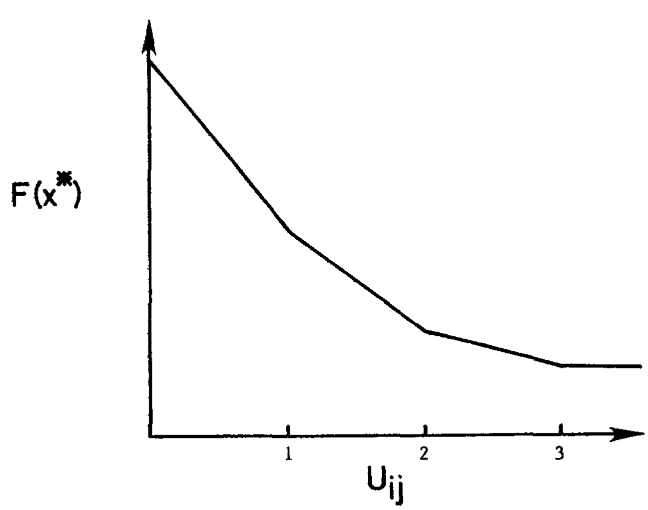

Fig. 2. Total optimal costs as a function of a single upper bound u_{ij}.

PROPOSITION 1 ([8], p. 160). *ν_{ij} is a subgradient of $F(x^*)$ with respect to the upper bounds u.* [*Note that in* [8], *the shadow price $\nu_{ij} = -(c_{ij} + \pi_j - \pi_i)$, since constraint 2.1b is written in the canonical form $-x \geq -u$, but we have chosen the form on 2.1b to give ν the cleaner interpretation as a subgradient.*]

What Proposition 1 establishes is that:

$$D_{ij}^{-} \leq c_{ij} + \pi_j - \pi_i \leq D_{ij}^{+}. \tag{2.5}$$

The problem is that we often use ν_{ij} as an approximation for D_{ij}^{-} or D_{ij}^{+}. For example, it is quite standard in introductory linear programming texts to state that a shadow price can be interpreted as the marginal value of another unit of the constraining resource. In the transportation context described earlier, the "resource" is the market demand from i to j, u_{ij}, and ν_{ij} then carries the interpretation of the marginal value of a unit of demand from it to j. As is described in Section 1, these shadow prices can be used to estimate marginal costs. The problem is that in general, $D_{ij}^{-} = \nu_{ij} = D_{ij}^{+}$ only if the solution is not degenerate, and it is widely recognized that most network problems are highly degenerate. In fact, it may be the case that $D_{ij}^{-} \ll D_{ij}^{+}$, implying that ν_{ij} may not be an accurate approximation of D_{ij}^{-} or D_{ij}^{+}. Numerical experiments illustrating this property are reported in Section 4.

Some authors (e.g. [6]) have attempted to mitigate problems of degeneracy by developing explicit algorithms for finding "non-degenerate" node duals π. For example, define the following perturbed problem:

$$(\mathrm{P}_i(\epsilon)) \quad \min_{x(\epsilon)} F_i(x(\epsilon)) = c^T x(\epsilon) \tag{2.6}$$

subject to:

$$Ax(\epsilon) = b + \epsilon e_i \tag{2.6a}$$

$$x(\epsilon) \leq u \tag{2.6b}$$

$$x(\epsilon) \geq 0 \tag{2.6c}$$

where e_i is a vector of zeroes with a one for the row corresponding to node i. Define directional derivatives γ_i^{+} and γ_i^{-} as we did above as follows:

$$\gamma_i^{+} = \lim_{\epsilon \downarrow 0} \frac{F_i(x^*(\epsilon)) - F(x^*)}{\epsilon} \tag{2.7}$$

and

$$\gamma_i^{-} = \lim_{\epsilon \uparrow 0} \frac{F_i(x^*(\epsilon)) - F(x^*)}{\epsilon}. \tag{2.8}$$

The plot of $F(x^*)$ with respect to the single right hand side variable b_i is also piecewise convex (as long as a feasible solution exists). The dual π_i is then a subgradient of $F(x^*)$ with respect to b_i and satisfies:

$$\gamma_i^{-} \leq \pi_i \leq \gamma_i^{+}. \tag{2.9}$$

This result is well known but does not appear to have been proven in print, so for completeness we state:

COROLLARY 1. *The vector of duals π is a subgradient of $F(x^*)$ with respect to the right hand side vector b.*

Proof. The proof easily parallels the proof of Proposition 1 in Kennington and Helgason.[9] Let:

$$z(b) = \mathrm{Min}_x\, c^T x = \mathrm{Max}_{\{\pi,\nu\}}\, \pi^T b + \nu^T u. \quad (2.10)$$

Let $\{\pi^*, \nu^*\}$ be the optimal duals for x^* and let $\{\bar{x}^*\}$ and $\{\bar{\pi}^*, \bar{\nu}^*\}$ be the optimal primal and dual solutions for a problem with right hand side $\bar{b}$. Then we must show:

$$z(b) - z(\bar{b})$$

$$= [(\pi^*)^T b + (\nu^*)^T u] - [(\bar{\pi}^*)^T \bar{b} + (\bar{\nu}^*)^T u] \quad (2.11a)$$

$$\geqslant [(\bar{\pi}^*)^T b + (\bar{\nu}^*)^T u] - [(\bar{\pi}^*)^T \bar{b} + (\bar{\nu}^*)^T u] \quad (2.11b)$$

$$\geqslant [(\bar{\pi}^*)^T (b - \bar{b})]. \quad (2.11c)$$

Equation 2.11b follows from Equation 2.11a since $\bar{\pi}^*$ and $\bar{\nu}^*$ are not necessarily optimal dual solutions for a problem with vector b. Equation 2.11c is the required result if $\bar{\pi}$ is a subgradient around the vector $\bar{b}$. Q.E.D.

The reason π is only a subgradient and not a gradient arises from degeneracy in the optimal basis. This degeneracy has an intuitively appealing interpretation. First we quickly summarize some well known properties of networks that can be found in any text on networks (see [6, 8, 11, 17]):

- Divide the basis matrix A into basic and nonbasic columns A_B and A_N, and assume the flow conservation constraint for the sink node S has been dropped as the redundant constraint. Then A_B^{-1} represents an arc-chain inidence matrix representing paths from each node in the network to the sink node S.
- Let c_B be the vector of costs for links in the basis. Then $\pi = c_B^T A_B^{-1}$ is a vector of path costs from each node to the root node S.

For example, Figure 3 shows a typical basis for a dynamic network. The corresponding matrix A_B is given by:

$$A_B = \begin{array}{c|ccccccccc}
 & 1\text{–}4 & 2\text{–}4 & 3\text{–}5 & 3\text{–}6 & 4\text{–}7 & 5\text{–}8 & 6\text{–}9 & 7\text{–}S & 9\text{–}S \\
\hline
1 & +1 & & & & & & & & \\
2 & & +1 & & & & & & & \\
3 & & & +1 & +1 & & & & & \\
4 & -1 & -1 & & & +1 & & & & \\
5 & & & -1 & & & +1 & & & \\
6 & & & & -1 & & & +1 & & \\
7 & & & & & -1 & & & & \\
8 & & & & & & -1 & & +1 & \\
9 & & & & & & & -1 & & +1
\end{array}$$

with inverse:

$$A_B^{-1} = \begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1\text{–}4 & +1 & & & & & & & & \\
2\text{–}4 & & +1 & & & & & & & \\
3\text{–}5 & & & & & -1 & & & -1 & \\
3\text{–}6 & & & +1 & & +1 & & & +1 & \\
4\text{–}7 & +1 & +1 & & +1 & & & & & \\
5\text{–}8 & & & & & & & & -1 & \\
6\text{–}9 & & & +1 & & +1 & +1 & & +1 & \\
7\text{–}S & +1 & +1 & & +1 & & & +1 & & \\
9\text{–}S & & & +1 & & +1 & +1 & & +1 & +1
\end{array}$$

The vector of duals π is easily calculated to be:

$$\pi = \begin{array}{c}1\\2\\3\\4\\5\\6\\7\\8\\9\end{array} \begin{bmatrix} +21 \\ +16 \\ +27 \\ +18 \\ +35 \\ +23 \\ +12 \\ +28 \\ +16 \end{bmatrix}.$$

Remember that in the context of the dynamic network shown in Figure 3, increasing, say, b_2 by one is equivalent to sending one more unit of flow from node 2 to the supersink S. It is clear from the figure that we cannot send one more unit of flow from 2 to S since the basic arc 2–4 is at its upper bound. Hence $\pi_2 = +16 < \gamma_2^+$, where the value of γ_2^+ is not obvious from the figure. Note, however, that we may push one more unit of flow from 1 to S, implying that in this case $\pi_1 = \gamma_1^+$. Similarly, $\pi_1 = +21 = \gamma_1^+$, but $\pi_1 > \gamma_1^-$ since the link from 1 to 4 is at its lower bound. Finally, $\gamma_8^- < \pi_8 = +28 < \gamma_8^+$, since the basis path from 8 to S has links at both upper and lower bounds.

FONG and SRINIVASAN[6] present the most comprehensive treatment of network degeneracy, presenting an algorithm for finding all "nondegenerate" duals π

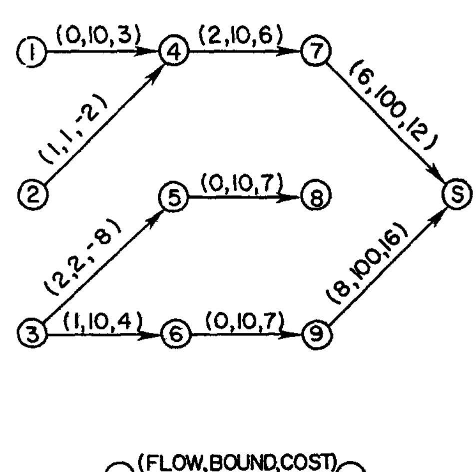

Fig. 3. Illustrative network basis.

in the context of a transportation problem. The important concept of this paper is to break the basis into a set of nondegenerate subtrees by dropping all links in the basis which are at their upper or lower bounds. These subtrees are then reconnected through the use of a specialized shortest path procedure. Unfortunately, the central ideas here, which are quite simple, are obscured by their use of rim operator vocabulary.

The important result in network degeneracy, which appears to be "generally well-known" but not widely reported within the network community is the relationship between least-cost flow augmenting paths and nondegenerate duals. Note that finding "nondegenerate" duals is misleading if the optimal solution is degenerate. We are really interested in finding certain directional derivatives with respect to the vector b. That is, if $\gamma_i^- < \gamma_i^+$, then we may find one basis that satisfies $\pi_i = \gamma_i^+$ and a different basis that satisfies $\pi_i = \gamma_i^-$, but we will obviously not be able to find a single basis that satisfies both conditions. Assume that we would like to find a basis, if it exists, that satisfies $\pi = \gamma^+$. The properties of the basis satisfies the following characterization theorem:

THEOREM 1 (Characterization theory for γ^+). *The basis that produces $\pi = \gamma^+$ is a least cost, flow augmenting tree from each node i to the sink node S.*

Proof. Let x^* be the optimal solution to P satisfying (2.1a)–(2.1c) and define the following optimization problem equivalent to P:

$$(\bar{\mathrm{P}})\quad \mathrm{Min}\quad c^T x - c^T x^* \tag{2.12}$$

subject to (2.6a) to (2.6c). Now define $z = x - x^*$ and rewrite (P) as:

$$\mathrm{Min}_z\ c^T z \tag{2.13}$$

subject to:
$$A(z + x^*) = b + e_i \tag{2.13a}$$
$$z + x^* \leq u \tag{2.13b}$$
$$z + x^* \geq 0. \tag{2.13c}$$

Since $Ax^* = b$, and if we let $u^+ = u - x^*$ and $u^- = x^*$, Equations 2.13a to 2.13c become:

$$Az = e_i \tag{2.14a}$$
$$z \leq u^+ \tag{2.14b}$$
$$z \geq -u^- \tag{2.14c}$$

where, again, e_i is a vector of zeroes with a 1 in the ith element. u^+ can be viewed as the amount by which flow may be increased on a link while u^- is the amount by which flow can be decreased. Let:

$$z = z^+ - z^-.$$

This is the equivalent to creating for each link (i, j) with flow z_{ij}^+ a mirror link (j, i) with flow z_{ij}^- and cost coefficient $c_{ji} = -c_{ij}$. We now wish to solve the following problem:

$$\mathrm{Min}\quad c^T z^+ - c^T z^- \tag{2.15}$$

subject to:
$$A(z^+ - z^-) = e_i \tag{2.15a}$$
$$z^+ \leq u^+ \tag{2.15b}$$
$$z^- \leq u^- \tag{2.15c}$$
$$z^+, z^- \geq 0. \tag{2.15d}$$

Problem (2.15)–(2.15d) is simply a shortest path problem over a modified network. Since $|e_i| = 1$, the flows will all be zero or one. Let G be the set of arcs and G^m be the set of mirrored arcs. Now let:

$$\bar{G} = \{i, j \mid (i, j) \in G, u_{ij}^+ > 0\}$$

and

$$\bar{G}^m = \{(j, i) \mid (j, i) \in G^m, u_{ji}^- > 0\}.$$

Thus $\bar{G}$ and $\bar{G}^m$ define the set of arcs with nonzero capacities. Let $\hat{z}$ be a vector of all link flows (including original and mirror links) and let $\hat{c}$ be the corresponding set of arc coefficients. Finally let $\hat{A}$ be the node-arc incidence matrix for the network associated with $\hat{z}$. We may now restate (2.15) to (2.15d) as:

$$\mathrm{Min}\quad \hat{c}^T \hat{z} \tag{2.16}$$

subject to:
$$\hat{A}\hat{z} = e_i \tag{2.16a}$$
$$\hat{z} \geq 0. \tag{2.16b}$$

Problem (2.16)–(2.16b) is clearly a standard shortest path problem whose optimal solution satisfies:

$$\hat{c}^T \hat{z}^* = \gamma_i^+. \tag{2.17}$$

Also, if $\hat{\pi}_i$ is the shadow price for the constraint for node i in (2.16a), then:

$$\hat{\pi}_i = \hat{c}^T \hat{z}^* \tag{2.18}$$

since $\hat{z}^*$ is the vector of positive link flows (all equal to one) from node i to the sink node S. Furthermore, it is easy to see that we may in a single shortest path calculation find the entire vector $\pi = \gamma^+$. Q.E.D.

The notion of least-cost, flow augmenting paths is quite old. BUSACKER and GOWEN[3] originally proposed solving the least-cost network flow problem by finding a sequence of least-cost flow augmenting paths. The out-of-kilter algorithm works by finding flow augmenting paths to adjust the kilter states of arcs. More recently, SCHNEIDER[19] proposes a homotopy procedure for solving least-cost network flow problems by finding flow augmenting paths between network subtrees. Jensen and Barnes[7] explicitly introduce their marginal network which is identical to problem (2.16)–(2.16b). The only step missing in this literature is pointing out the direct relationship between solving least-cost flow augmenting problems and network duals. This long history makes it difficult to claim that Theorem 1 presents anything new, but this statement of the results presents, without ambiguity, the characterization of the duals $\pi = \gamma^+$ as flow augmenting paths from the root node to each node of the network.

It is of course clear from this discussion that we can also find the duals $\pi = \gamma^-$ by solving (2.16) with the right hand side of 2.16a multiplied by minus one. The problem is an exact mirror of finding $\pi = \gamma^+$. Note that for both problems it is possible that the network problem may be infeasible for a particular perturbation. The corresponding shortest path problem 2.16 will simply not find a path to one or more nodes resulting in γ_i^+ (or γ_i^-) $= \infty$ for those nodes i where no augmenting path can be found to (or from) the sink node.

Let us now return to our original question of finding the sensitivity of $F(x)$ with respect to a given upper bound. Assume we wish to find D_{ij}^+, the effect of increasing u_{ij} by 1. We know that $D_{ij}^+ \geq c_{ij} + \pi_j - \pi_i$. We have already characterized the property of $\pi_i = \gamma_i^+$ as a flow augmenting path from i to the supersink. We can now characterize D_{ij}^+ as follows:

COROLLARY 2 (Characterization theorem for D_{ij}^+). *D_{ij}^+ is given by:*

$$D_{ij}^+ = \mathrm{Min}\{l_{ji} + c_{ij}, 0\}$$

where l_{ji} is the length of a least-cost flow augmenting path from node j back to node i.

Proof. This is a direct corollary of Theorem 1. If $x_{ij}^* < u_{ij}$, then $D_{ij}^+ = 0$. Assume $x_{ij}^* = u_{ij}$ and now assume that the optimal solution of the perturbed problem $\bar{x}^*$ satisfies $\bar{x}_{ij}^* = \bar{u}_{ij} = u_{ij} + 1$. Now modify $\bar{P}$ by defining $\hat{x}_{ij} = \bar{x}_{ij} - 1$ and $\hat{x}_{kl} = \bar{x}_{kl}$ for $(k, l) \neq (i, j)$. This standard transformation is equivalent to reducing $\bar{u}_{ij}$ by one (thus $\bar{u}_{ij}$ is now equal to the original upper bound), and defining a new right hand side vector b using:

$$\hat{b}_n = \begin{cases} b_i - 1 & n = i \\ b_j + 1 & n = j \\ b_k & k \neq i, j \end{cases}. \tag{2.19}$$

Defining the variable $\hat{z}$ as done previously (that is, including mirrored links and eliminating all links with zero bounds), we find ourselves solving:

$$\mathrm{Min} \quad \hat{c}^T\hat{z} = c_{ij} \tag{2.20}$$

subject to:

$$\hat{A}\hat{z} = e_j - e_i \tag{2.20a}$$

$$\hat{z} \geq 0. \tag{2.20b}$$

This is equivalent to (2.16)–(2.16b) except that the right hand side now has $a + 1$ and $a - 1$, implying that we are solving a least-cost flow augmenting path from node j back to node i. Let l_{ji} be the length of this path. Then:

$$D_{ij}^+ = \mathrm{Min}\{l_{ji} + c_{ij}, 0\}. \tag{2.21}$$

If $l_{ji} + c_{ij} > 0$, then costs would rise by increasing the flow along this link, implying the optimal solution would not actually change.

As before, the primary objective here is to explicitly characterize the directional derivatives D_{ij}^+ and D_{ij}^- as flow augmenting paths (D_{ij}^- is found as a flow augmenting path from i to j). Note that we can find the entire vector γ^+ (or γ^-) using a single shortest path calculation from all nodes i into the supersink. Similarly, we can calculate D_{ij}^+ for all nodes i using a single shortest path calculation, but this would have to be repeated for each node j. The work required to do this is the same as efficiently reoptimizing $F(x)$ for each individual perturbation of u_{ij}.

The next section presents a much more efficient method for accurately approximating D_{ij}^+ and D_{ij}^- without the need to reoptimize $F(x)$ for each possible perturbation of u_{ij}, and also without the need to store D_{ij}^+ or D_{ij}^- explicitly for each link (i, j).

3. AN EFFICIENT PROCEDURE FOR APPROXIMATING D_{ij}

RETURN first to the reduced cost calculation:

$$\bar{c}_{ij} = c_{ij} + \pi_j - \pi_i. \tag{3.1}$$

The appeal of this calculation is that it allows us to estimate the impact of increasing the upper bound on any link (i, j) using a single node length vector of π. We might hope that we could use our ability to find "nondegenerate" duals, such as $\pi = \gamma^+$, to refine (3.1). The following proposition suggests that this approach will never work consistently:

PROPOSITION 2.

i) In general, $\bar{c}_{ij} = D_{ij}^+$ if and only if the network basis is not degenerate.

ii) For a given optimal solution x^*, if one optimal basis is degenerate for a network then all optimal bases are degenerate.

iii) For a network with a degenerate basis, there does not in general exist any single vector π such that $\bar{c}_{ij} = D_{ij}^+$ for all links (i, j).

Proof. Part i of the proposition is proved trivially. Note that it is possible that $\bar{c}_{ij} = D_{ij}^+$ for a particular link (i, j) through a suitable choice of link costs, but it will not in general be true for all links and all networks. Part ii is seen by noting that since all links between upper and lower bounds must be in the basis, if there is at least one optimal basis that is degenerate, then the links between their upper and lower bounds do not form a tree, and therefore all bases must be degenerate. Finally, we can prove part iii by presenting a single example of a network where this occurs. Such a network is shown in Figure 4a, where there are only two possible optimal bases. Let D^+_a be the impact of increasing the upper bound on arc a, where $a = \{1, 2, 3, 4\}$. Note that $D^+_1 = 0$ and $D^+_4 = 0$. If the basis corresponds to Figure 4b, $\bar{c}_i = D^+_1 = 0$, but $\bar{c}_4 = -7 \neq D^+_4$. Similarly, if the basis is as shown in Figure 4c, $\bar{c}_4 = D^+_4 = 0$, but $\bar{c}_1 = -7 \neq D^+_1$. Q.E.D.

Of course, Proposition 2 leaves unanswered the question of how accurately $\bar{c}_{ij}$ can be calculated using a single vector π. This empirical question is addressed later.

The simple reduced cost calculation in Equation 3.1 can be dramatically improved using the following intuitive argument. We can think of raising an upper bound u_{ij} by 1 and assuming the optimal flow increases by one as pushing one more unit of flow from i to j, plus the cost of moving a unit of flow out of j plus the cost of moving a unit of flow into i. This cost of moving a unit of flow out of j can be approximated as the cost of moving one more unit of flow from the node j to the root node S. Thus we want the flow augmenting path from j to the root node S given by γ^+_j. Similarly, the cost of moving a unit of flow into node i can be approximated using the flow augmenting path from the root node to node i, but since we are effectively decreasing b_i, this cost is given by $-\gamma^-_i$.

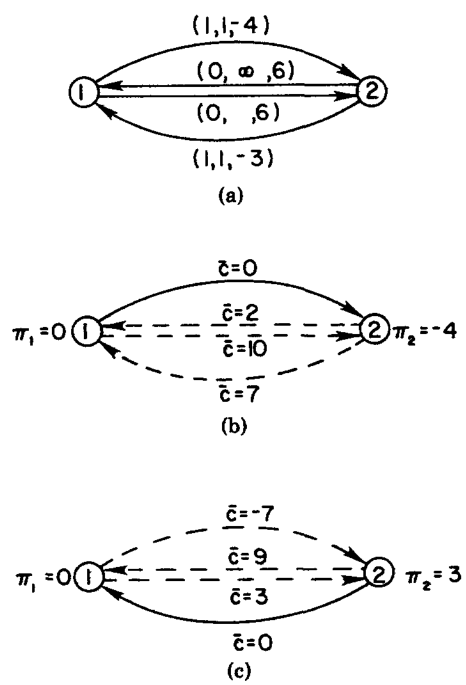

Fig. 4. Illustration of alternative network bases.

Thus our new approximate reduced cost calculation is:

$$D^+_{ij} \approx \hat{D}^+_{ij} = \min\{c_{ij} + \gamma^+_j - \gamma^-_i, 0\}. \tag{3.2}$$

By simply reversing the logic, we immediately obtain an approximation for D^-_{ij}:

$$\begin{aligned} D^-_{ij} &\approx \min\{0, c_{ij} + \gamma^-_j - \gamma^+_i\} && \text{if } x^*_{ij} = u_{ij} \\ &= 0 && \text{if } x^*_{ij} < u_{ij}. \end{aligned} \tag{3.3}$$

Equations 3.2 and 3.3 are simple, elegant calculations for the directional derivatives of $F(x^*)$ with respect to u_{ij}, using two easily calculated node length vectors instead of one. The challenge now is to understand the theoretical properties of the approximations and then to evaluate them empirically.

Assume that for the modified problem we have $\bar{x}^*_{ij} = x^*_{ij} + 1$, and make the transformation, as we did earlier, $\hat{x}^*_{ij} = \bar{x}^*_{ij} - 1 = x^*_{ij}$. This creates a new right hand side vector $\hat{b}$ as given by (2.19), and a new vector of upper bounds $\hat{u}$ where $\hat{u}_{ij} = \bar{u}_{ij} - 1 = u_{ij}$. Using this transformation and starting with (2.12) we wish to solve:

$$\text{Min} \quad c^T\hat{x} - c^T x^* + c_{ij} \tag{3.4}$$

subject to:

$$A\hat{x} = \hat{b} \tag{3.4a}$$

$$\hat{x} \leqslant \hat{u} \tag{3.4b}$$

$$\hat{x} \geqslant 0. \tag{3.4c}$$

As before, let $z = \hat{x} - x^*$ producing:

$$\text{Min}_z \; F(z) = c^T z + c_{ij} \tag{3.5}$$

subject to:

$$Az = e_j - e_i \tag{3.5a}$$

$$z \leqslant u^+ \tag{3.5b}$$

$$z \geqslant u^-. \tag{3.5c}$$

We have already demonstrated that (3.5) could be solved by defining $z = z^+ - z^-$ and using the notion of mirrored arcs. It is more convenient to work directly in terms of z which is allowed to be negative. Now consider the following two problems:

$$\text{Min}_{z(j)} \; F^+_j(z) = c^T z(j) \tag{3.6}$$

subject to:

$$Az(j) = e_j$$

$$z(j) \leqslant u^+$$

$$z(j) \geqslant u^-.$$

and

$$\text{Min}_{z(i)} \; F^-_i(z) = c^T z(i) \tag{3.7}$$

subject to:

$$Az(i) = -e_i$$

$$z(i) \leqslant u^+$$

$$z(i) \geqslant u^-.$$

Clearly, $\gamma_j^+ = F_j^+(z^*(j))$ and $\gamma_i^- = -F_i^-(z^*(i))$. Now define:

$$z^*(i, j) = z^*(j) + z^*(i) \tag{3.8}$$

where $z(i, j)$ is an element by element sum of the vectors $z(i)$ and $z(j)$. Ideally we would like to find that $z^* = z^*(i, j)$, since this would imply that:

$$\begin{aligned} F_{ij}(z^*) &= c^T z + c_{ij} \\ &= c^T z^*(i) + c^T z^*(j) + c_{ij} \\ &= F_i(z^*(i)) + F_j(z^*(j)) + c_{ij}. \end{aligned} \tag{3.9}$$

Since $F_{ij}(z^*)$ forces an additional unit of flow from i to j,

$$D_{ij}^+ = \mathrm{Min}\{F_{ij}(z^*), 0\}. \tag{3.10}$$

The limitation of this result is that since we do not know z^*, we cannot check if $z^* = z^*(i, j)$. We can, however, infer some of the properties of $\hat{D}_{ij}^+$.

PROPOSITION 3.

$$\hat{D}_{ij}^+ \geqslant D_{ij}^+. \tag{3.11}$$

Proof. As before, let l_{ji} be the cost of a least-cost flow augmenting path from j to i. Then $D_{ij}^+ = c_{ij} + l_{ji}$. Alternatively, $\hat{D}_{ij}^+ = c_{ij} + l_{js} + l_{sj}$ (S is the supersink) where clearly $l_{js} + l_{si} \geqslant l_{ji}$. Q.E.D.

We now have a bound for D_{ij}^+ by using

$$0 \geqslant \hat{D}_{ij}^+ \geqslant D_{ij}^+ \geqslant \bar{c}_{ij}. \tag{3.12}$$

We show numerically in Section 4 that $\hat{D}^+$ is a tight bound for D^+, while $\bar{c}$ can be a loose lower bound.

The bound in 3.11 arises because the approximation is a restriction on the original problem. Interestingly, the approximation can also be viewed as a relaxation, suggesting that we might find $\hat{D}_{ij}^+ < D_{ij}^+$. To see this, note that it is always possible to find vectors z_1 and z_2 satisfying

$$z_1 + z_2 = z^* \tag{3.13a}$$

$$Az_1 = e_j \tag{3.13b}$$

$$Az_2 = -e_i. \tag{3.13c}$$

For this reason we may rewrite (3.5) as

$$\min_{z_1, z_2} c^+(z_1 + z_2) \tag{3.14}$$

subject to (3.13b), (3.13c) and

$$z_1 + z_2 \leqslant u^+ \tag{3.14a}$$

$$z_1 + z_2 \geqslant u^-. \tag{3.14b}$$

If we replace (3.14a–b) with

$$z_1 \leqslant u^+ \tag{3.15a}$$

$$z_2 \leqslant u^+ \tag{3.15b}$$

$$z_1 \geqslant u^- \tag{3.15c}$$

$$z_2 \geqslant u^- \tag{3.15d}$$

then 3.14 decomposes into (3.6) and (3.7). Constraints (3.15a) and (3.15b) represent a relaxation of 3.14a since they allow $z_1 + z_2 \geqslant u^+$. However, it can be shown this will never occur, since it implies the presence of a negative cycle which violates the assumption that z^* solves 3.5.

4. NUMERICAL EXPERIMENTS

THE NEW approximation was evaluated using two types of networks that have been used for solving dynamic vehicle allocation problems for truckload motor carriers. The first network used, denoted NET1, is a network based on a stochastic programming heuristic suggested in Powell,[14] illustrated in Figure 5. This network structure is actively being used in practice to solve DVA problems for motor carriers because of its ability to handle forecasting uncertainties. From the perspective of sensitivity analyses, the major characteristic of this network is that it is relatively shallow, often involving only 4 to 10 links between nodes in the first time period and the supersink. The second network used, denoted NET2, is like that illustrated in Figure 1. Here, a network was generated covering 14 days, with loaded movement links having an upper bound of one scattered throughout the planning horizon. Empty movement links and inventory links are

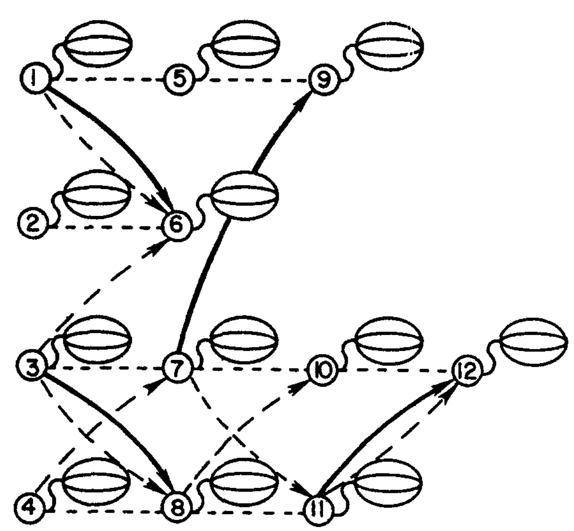

Fig. 5. Illustration of network resulting from stochastic programming heuristic.

allowed throughout the planning horizon. The characteristics of each network, and some information about the nature of the basis in the original optimal solution, are summarized in Table I. The percent of bounded links, and the percent of degenerate basic links (the percent of links in the basis at their upper or lower bounds) provide a measure of the sparsity of the basic network and the associated optimal basis.

For each network, we compared the shadow price on the upper bounds of selected loaded movement links in the first and second time periods. For NET1, these links corresponded to loads that are already known to the carrier, since forecasted loads are handled through the stochastic link clusters. For NET2, the loaded movement links represented both the "known loads" and forecasted loads. For both networks, we only considered the shadow prices on loaded movement links that were classified as "known loads." Both networks were generated from the same set of known loads and the same set of forecasts, drawn from the actual data of a major motor carrier. After calculating the standard reduced cost from the simplex duals, and the new reduced cost based on the augmenting path approximation, we then, in a brute force fashion, raised the upper bound of each link, in sequence, and determined the actual change in the objective function. Links whose flow was zero in the original optimal solution were ignored, since raising these upper bounds would have no effect. The relative error was calculated for each case by comparing the errors for each shadow price relative to the exact change in costs.

The results of these experiments are summarized in Table II, A and B, divided into ranges of relative errors. For NET1, 18% of the shadow prices exhibited errors of 20% and 9% showed errors in excess of 50%. The fact that 70% showed errors less than 10% is actually quite good, but not good enough for the kinds of applications discussed in Section 1. With the flow augmenting approximation, on the other hand, 87% of the results were exact (versus 24% for the simplex duals), and there were no errors greater than 5%.

For the deeper network NET2, the results were quite similar but even more favorable for the new approximation. In this case, fully 99% of the estimated cost changes for the flow augmenting path approximation were exact, versus 39% for the simplex duals. Two out of the 159 observations exhibited errors between 5 and 10%, while again 9% of the simplex reduced costs exhibited errors greater than 50%.

TABLE I

Characteristics of the Test Networks

Network	No. of Regions	Time Periods	No. of Links	Percent of Bounded Links	Links in Basis	Percent of Degenerate Links
NET1	29	5[a]	8548	30	1056	41
NET2	29	14	10733	38	1514	81

[a] Network average. Each region is represented by a varying number of time periods depending on the data, with a minimum of two time periods for each region.

5. CONCLUSIONS

BASED ON the theoretical foundation and the supporting numerical experiments, we conclude that the flow augmenting path approximation for calculating reduced costs on upper bounds for links is a simple and highly accurate method for estimating the cost impact from raising (or lowering) the upper bound on a link. The numerical experiments are limited to dynamic networks typical of those arising in dynamic

TABLE II

Simplex Duals versus New Approximation for Networks

Network	Cumulative Error	Simplex Duals			New Approximation		
		Total	Percent	Cumulative	Total	Percent	Cumulative
NET1	0	38	24	24	138	87	87
	>0–1	11	7	31	5	3	91
	>1–5	34	22	53	15	9	100
	>5–10	27	17	70	0	0	100
	>10–15	11	7	77	0	0	100
	>15–20	8	5	82	0	0	100
	>20–50	14	9	91	0	0	100
	>50	15	9	100	0	0	100
NET2	0	62	39	39	157	99	99
	>0–1	8	5	44	0	0	99
	>1–5	33	21	65	0	0	99
	>5–10	12	7	72	2	1	100
	>10–15	11	7	79	0	0	100
	>15–20	6	4	83	0	0	100
	>20–50	13	8	91	0	0	100
	>50	14	9	100	0	0	100

vehicle allocation problems in truckload trucking, and it is possible that these networks are particularly well suited to the approximation. It is most likely that for some applications that the underlying network will not be as degenerate as the ones used here, implying that the simplex duals may be more accurate in other instances. However, we are quite sure that the new approximation will always, on an aggregate population of links, outperform the estimates of reduced costs produced using standard simplex calculations. Most important, the accuracy of the new approximation is such that the results may be used in real-time applications where unsophisticated users must use these results. It is in these applications that even a 3 to 5% rate of errors over 50% is unacceptable since they produce a loss in confidence in the underlying model.

A separate issue that has not been addressed in this paper is the applicability of the results to static networks. The key characteristic of dynamic networks is the presence of a supersink which serves as a natural root node for the flow augmenting paths. We do not have to use the supersink as this root node, but it is not clear whether we would obtain the same empirical performance as that reported in Section 4. If the procedure is robust relative to the choice of root node, then the process should work for classical static networks.

Section 1 motivated the use of these accurate shadow prices for a class of problems arising in truckload trucking. Accurate sensitivity results, however, may be useful in other areas of subgradient optimization where shadow prices are used. A powerful integer programming heuristic for fixed charge networks is to sequentially add links where the flow and the upper bound are currently set to zero but where the reduced cost on the upper bound suggests that savings would occur if the link is added. Subgradient algorithms used within a resource directive decomposition strategy for capacitated multicommodity flow problems may also benefit. In this case, the upper bound u_{ij} for a link is divided into separate bounds u_{ij}^k where $u_{ij} = \sum_k u_{ij}^k$. Adjustments are made to the vector of allocations $\{u^k, k = 1, \ldots, C\}$, where C is the number of commodities, based on the reduced costs of these constraints in the separate single commodity subproblems. Note, however, that an accurate estimate of the reduced cost is only valuable if you know in advance whether you are trying to increase or decrease an upper bound. Thus if $u_{ij}^k = 0$, then we know in advance that we are interested in trying to increase the allocation for commodity k, with an estimated reduced cost of $\hat{D}_{ij}^+$. Correspondingly, if $u_{ij}^k = u_{ij}$, then we know we are trying to decrease the allocation for commodity k, with an estimated reduced cost of $\hat{D}_{ij}^-$. If $0 < u_{ij}^k < u_{ij}$, then we have no a priori basis to believe that we are trying to increase or decrease the capacity allocation for commodity k.

APPENDIX: AN ALGORITHM FOR FINDING LEAST-COST FLOW AUGMENTING PATHS

FINDING least-cost flow augmenting paths is a straightforward shortest path problem over an augmented network, as pointed out earlier (and has been observed by many other authors). Alternatively, some authors[6] have suggested solving shortest path problems between subtrees formed in the basis of links between their upper and lower bounds. Since these links are always in any basis, it seems natural to use this output of the optimization as a kind of "advanced start" for finding least-cost, flow augmenting paths. It is not clear how much such a procedure helps, given the extent to which network bases tend to be degenerate (dynamic networks used by the authors have been found to exhibit 40–80% of the links in the basis at their upper or lower bound).

Following is the description of a standard algorithm for finding flow augmenting paths used which is easy to implement, does not require forming an augmented network and has proved to be exceptionally fast in practice. The algorithm is provided here only for completeness and does not present anything new.

We first define the following:

$x_{ij}^* =$ optimal flow on link (i, j)
$c_{ij} =$ cost coefficient for link (i, j)
$u_{ij} =$ upper bound for link (i, j)
$G_i = \{j \mid (i, j) \in G\}$
$H_j = \{i \mid (i, j) \in G\}$
$\pi_i =$ flow augmenting dual
$p_i =$ successor node on the least-cost flow augmenting path node i to the supersink s
$l_i =$ successor link on the least-cost flow augmenting path node i to the supersink s
$l(i, j) =$ link number from i to j.

LIST(i)
- $= 0$ if node i has never been reached,
- $= j$ if node i is currently in the list, and node j is the next node in the list,
- $= M$ if node i is currently in the list, but is at the end of the list,
- $= -1$ if node i was in the list but is no longer.

The main requirement is that the data structures be available to quickly identify the sets G_i and H_j. For example, if the network is stored in forward star form (see, for example, Dial et al.[5]), then the set of all links leaving node i, G_i, are easily found using forward star pointers (which point to the first link out of each node). To find the set of links coming into a node, it is necessary to form a linked list that uses a node length vector to point to the first link coming into a node j. Then a link length vector is used to point from one link coming into node j to the next link coming into node j.

With this data structure in mind, the algorithm used to find the flow augmenting duals is listed below. The procedure is a direct modification of Pape's shortest path algorithm (see [5]), except that from a given node n, we look first for links entering node n which are below their upper bound, implying that we may increase flow over this link, and then we look for links leaving node n which are above their lower bound, over which we would be decreasing flow.

Step 0. Initialization:
  For all nodes $i \in N$, set $\pi_i = M$, $p_i = 0$, $l_i = 0$ and LIST$(i) = 0$.
  For the root node s, let $\pi_s = 0$, LIST$(s) = M$, $n = s$ and LAST $= s$.

Step 1. For each $i \in H_n$ do:
  If $x^*_{in} < u_{in}$ then:
    If $\pi_n + c_{in} < \pi_i$ then:
      Set: $\pi_i = \pi_n + c_{in}$
        $\pi_i = n$
        $l_i = l(i, j)$
        if LIST$(i) = 0$ then:
          LIST(LAST) $= i$
          LAST $= i$
          LIST$(i) = M$
      Else if LIST$(i) < 0$ then:
        LIST(i) = LIST(n)
        LIST$(n) = i$
      Endif;
    End;
  End;

Step 2. For each $j \in G_n$ do;
  If $x^*_{nj} > 0$ then:
    If $\pi_n - c_{nj} < \pi_j$ then:
      Set: $\pi_j = \pi_n - c_{nj}$
        $p_j = n$
        $l_j = l(n, j)$
        If LIST$(j) = 0$ then:
          LIST(LAST) $= j$
          LAST $= j$
          LIST$(j) = M$
        Else if LIST$(j) < 0$ then:
          LIST(j) = LIST(n)
          LIST$(n) = j$
      Endif;
    End;
  End;

Step 3. Set $m = n$. If LIST$(n) = M$, go to Step 4. Otherwise, set n = LIST(n), LIST$(m) = -1$ and go to Step 1.

Step 4. Set $\gamma^+ = \pi$.

The above procedure finds the least-cost flow augmenting path from each node to the supersink node s. A parallel algorithm, not shown here, is easily developed for finding least-cost flow decreasing paths into the supersink.

As mentioned above, this algorithm does not take advantage of any information from the optimal basis which might be used to improve the performance. However, the algorithm as it stands is particularly easy to implement and appears to be exceptionally fast. For example, experience with the procedure showed that execution times for a 10,700 link network were approximately 2 CPU seconds on a Microvax II. This speed arises principally because we are solving nothing more than a shortest path problem. The original optimization of the network, using an efficient network simplex code with an advanced start routine, required 47 CPU seconds. The real savings in CPU time, of course, derives from the fact that we do not need to perform any additional reoptimizations to determine accurate reduced costs once the vectors γ^+ and γ^- are found.

ACKNOWLEDGMENTS

THIS WORK was performed while the author was on sabbatical at the Centre de recherche sur les transports, Université de Montréal. He warmly appreciates the opportunity provided by the CRT to conduct this work, and the many conversations he had with the members of the CRT on this subject. The computational work was performed by David J. Cape. This research was supported by a Presidential Young Investigator Award from the National Science Foundation, grant ECE-8451466.

REFERENCES

1. W. P. ALLMAN, "An Optimization Approach to Freight Car Allocation under Time-Mileage per Diem Rental Rates," *Mgmt. Sci.* **18,** 10, 567–574 (1972).
2. J. H. BOOKBINDER AND S. P. SETHI, "The Dynamic Transportation Problem: A Survey," *Naval Res. Logist. Quart.* **27,** 1, 65–87 (1980).
3. R. G. BUSACKER AND P. J. GOWEN, "A Procedure for Determining a Family of Minimal Cost Network Flow Patterns," ORO Technical Report 15, Johns Hopkins University, 1961.
4. P. J. DEJAX AND T. G. CRAINIC, "A Review of Empty Flows and Fleet Management Models in Freight Transportation," *Trans. Sci.* **21,** 227–247 (1987).
5. R. DIAL, F. GLOVER, D. KARNEY AND D. KLINGMAN, "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees," *Networks* **9,** 215–248 (1979).
6. C. O. FONG AND V. SRINIVASAN, "Determining All Nondegenerate Shadow Prices for the Transportation Problem," *Trans. Sci.* **11,** 199–222 (1977).

7. P. A. JENSEN AND J. W. BARNES, *Network Flow Programming*, John Wiley & Sons, New York, 1980.
8. W. C. JORDAN AND M. A. TURNQUIST, "A Stochastic, Dynamic Model for Railroad Car Distribution," *Trans. Sci.* **17**, 123–145 (1983).
9. J. L. KENNINGTON AND R. V. HELGASON, *Algorithms for Network Programming*, John Wiley & Sons, New York, 1980.
10. S. C. MISRA, Linear Programming of Empty Wagon Disposition," *Rail Internat.* **3**, 151–158 (1972).
11. G. P. OUIMET, "Empty Freight Car Distribution," Master's thesis, Queen's University, Kingston, Ontario, 1972.
12. D. T. PHILLIPS AND A. GARCIA-DIAZ, *Fundamentals of Network Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
13. W. B. POWELL, Y. SHEFFI AND S. THIRIEZ, "The Dynamic Vehicle Allocation Problem with Uncertain Demands," in *Ninth International Symposium on Transportation and Traffic Theory*, pp. 357–374, J. Vormuller and R. Hamerslag (eds.), VNU Science Press, The Netherlands, 1984.
14. W. B. POWELL, "A Stochastic Model of the Dynamic Vehicle Allocation Problem," *Trans. Sci.* **20**, 117–129 (1986).
15. W. B. POWELL, "An Operational Planning Model for the Dynamic Vehicle Allocation Problem with Uncertain Demands," *Trans. Res.* **21B**, 217–232 (1987).
16. W. B. POWELL, Y. SHEFFI, K. S. NICKERSON, K. BUTTERBAUGH AND S. ATHERTON, "Maximizing Profits for Truckload Motor Carriers: A New Framework for Pricing and Operations," *Interfaces* **18**, 21–41 (1988).
17. W. B. POWELL, "A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem," in *Vehicle Routing: Methods and Studies*, B. Golden and A. A. Assad, eds., North-Holland, New York, 1988.
18. R. T. ROCKAFELLAR, *Network Flows and Monotropic Optimization*, John Wiley & Sons, New York, 1984.
19. M. H. SCHNEIDER, "A Complementary Pivoting Algorithm for Linear Network Problems," *Ann. Opns. Res.* **5**, 439–462 (1985).
20. Y.-S. SHAN, "A Dynamic Multicommodity Network Flow Model for Real Time Optimal Rail Freight Car Management," Ph.D. dissertation, Department of Civil Engineering, Princeton University, 1985.
21. W. W. WHITE, "Dynamic Transshipment Networks: An Algorithm and its Application to the Distribution of Empty Containers," *Networks* **2**, 211–236 (1972).
22. W. W. WHITE AND A. M. BOMBERAULT, "A Network Algorithm for Empty Freight Car Allocation," *IBM Syst. J.* **8**, 147–169 (1970).

(Received, June 1988; revisions received March and May 1989; accepted May 1989)