MANAGEMENT SCIENCE
Vol. 50, No. 3, March 2004, pp. 309–325
ISSN 0025-1909 | EISSN 1526-5501 | 04 | 5003 | 0309
DOI 10.1287/mnsc.1030.0161

Long-Distance Access Network Design

Rosemary T. Berger
Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, Pennsylvania 18015, rosemary.berger@lehigh.edu

S. Raghavan
Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742, raghavan@umd.edu

Long-distance telephone companies in the United States pay access fees to local telephone companies to transport calls that originate and terminate on their networks. These charges form the largest portion of the cost of providing long-distance service. Recent changes in the structure of access rates, which were mandated by the Federal Communications Commission (FCC), have created opportunities for long-distance companies to better manage access costs. In this paper, we develop an optimization-based approach to the economic design of access networks. Our novel solution approach combines stochastic aspects of the problem with a challenging discrete facility location problem in a three-phase algorithm. Computational results indicate a potential cost savings of hundreds of millions of dollars annually for long-distance companies.

Key words: integer programming; hub location; queueing; network flows; telecommunications
History: Accepted by Thomas M. Liebling, former department editor; received August 9, 2002. This paper was with the authors 5 months for 1 revision.

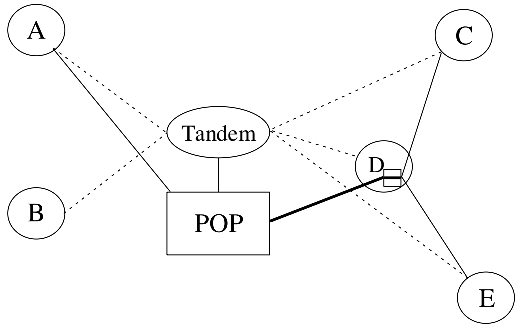
1. Introduction

As is evident from the fierce price wars of the 1990s and the continued attempts by the Baby Bells to obtain regulatory approval to offer service, the long-distance communications industry in the United States is competitive. In the two decades since the breakup of the Bell Telephone System, the price of a long-distance phone call has fallen from more than $0.32 per minute in 1984 ($0.52 per minute in 2000 dollars) to an average of $0.12 per minute in 2000 (FCC 2001). And there are more than 700 companies offering long-distance service, all vying to capture a larger share of the $109 billion market (Lande and Lynch 2002), dominated by AT&T, WorldCom, and Sprint.

To compete effectively in the market, particularly given the difficult economic climate of the past few years, both large and small companies must explore every available avenue for reducing operating expenses. Access charges, which are the fees long-distance companies pay to local telephone companies to originate and terminate calls, comprise the largest portion of a long-distance company's network operating expenses. Traditionally, access rates have been set higher than the actual cost to the local telephone company as a means of indirectly subsidizing local phone service. In recent years, as part of its efforts to open local markets to competition, the FCC adopted a market-based approach to drive access charges toward the cost of providing service (FCC 1999). The resulting changes in the access rate structure have had two main impacts on the long-distance industry. One, the per-minute costs for originating and terminating a call have decreased, as a percentage of revenue, from 35% in 1996 to 25% in 2000. (At the time of divestiture in 1984, per-minute access charges accounted for 54% of revenue (FCC 2001).) Two, the relative price changes among the cost elements for the two main alternatives for interconnecting with the local telephone company have created opportunities for the long-distance companies to achieve additional savings by optimizing their access strategy. We call the problem of designing networks that minimize the access charges paid by a long-distance company to the local telephone company the long-distance access network design (LDAND) problem. In this paper, we develop and test an optimization-based solution approach for the LDAND problem.

The LDAND problem arises in each of the 184 local access and transport areas (LATAs) in the U.S. local telephone companies handle calls within a LATA, while long-distance companies handle calls between LATAs. Figure 1 illustrates the key components of the LDAND problem. (More detailed descriptions of access networks are available on the websites of several telephone companies. See, for example, www.qwest.com.) The LDAND problem involves two types of nodes: end offices (EOs), labeled A, B, C, D, and E in Figure 1, and a point of presence (POP). An EO is a collection point in the local telephone

Figure 1 Access Network with TST, DTT, and a Hub



company's network for many (hundreds) individual customers. A long-distance carrier interconnects its facilities with the local company's network at a specially designated node, the POP. Transmission services leased from the local telephone company carry long-distance traffic between the POP and the EO that serves an individual customer.

Local telephone companies currently offer two types of transmission service between an EO and a POP, and they specify the associated charges in a tariff document filed with the FCC. Table 1 shows a sample tariff table. With *direct trunked transport* (DTT), a long-distance company leases dedicated facilities from the local telephone company to carry traffic between an EO and the POP. The long-distance company pays the local company a monthly fixed charge based on the type and the length of the facility leased. In Figure 1, dedicated DTT facilities leased between EO-A and the POP are represented by the solid line connecting the two nodes. With *tandem switched transport* (TST), the local company carries traffic on shared circuits from an EO through a tandem switch to the POP. The long-distance company pays fees to the local company based on usage—a per-minute charge for each minute of traffic carried. In Figure 1, shared TST circuits between EO-B and the tandem switch are represented by the dotted line connecting the two nodes.

Each type of transport service favors a different traffic pattern. With DTT, a long-distance company must pay the cost to lease all of a facility's capacity regardless of how much traffic is carried. Consequently, leasing dedicated facilities is generally cost effective for EOs generating and receiving large amounts of traffic at a steady rate throughout the day. Conversely, with TST, the long-distance company only pays for the actual traffic carried, so there is no penalty associated with low utilization. Compared to the equivalent per-minute cost on a dedicated facility with high utilization, however, the per-minute charge for TST is more expensive; therefore, TST is cost effective for EOs with small amounts of traffic occurring infrequently throughout the day.

Table 1 Example Tariff Table

	Fixed cost ($)	Cost per mile ($)
DS1	85.00	20.00
DS3	362.50	81.50
TST (per minute)	0.001849	0.000032
Multiplexer	623.00	—

When permitted, the most economical solution is to lease a combination of the two service types and to use a "fixed alternate routing" policy (Freeman 1996), as illustrated for EO-A in Figure 1. Under this strategy, traffic from (to) EO-A would be offered first to the group of dedicated DTT circuits connecting EO-A and the POP. Any traffic blocked by the dedicated circuits would overflow to the shared circuits (dotted line from EO-A to the tandem switch) and be transported to (from) the POP via the tandem switch. Determining the number of dedicated circuits to lease between each EO and the POP is a dimensioning problem, and the solution depends on the distribution of the traffic throughout the day and the costs for DTT and TST.

For access networks, dedicated transmission facilities are available at two standard rates: 1.544 Mbps and 44.736 Mbps, labeled DS1 and DS3, respectively. In general, a long-distance company will lease DS1 facilities for DTT. However, because of strong economies of scale, DTT costs often can be reduced by creating hubs. Under a hubbing strategy, each EO in a cluster is connected by DS1s to a hub node (which may be the POP). At the hub, their multiple low-speed DS1 signals are consolidated into a high-speed DS3 signal. The signals are consolidated using an electronic device called a multiplexer, which has the capacity to combine 28 DS1 signals into one DS3 signal. The DS3 signal then is transmitted from the hub to the POP. In Figure 1, a multiplexer is installed at EO-D to create a hub. The DS1s from EO-C, EO-D, and EO-E all are consolidated into a DS3 signal that connects EO-D to the POP. A second hub would be located at the POP to multiplex the DS1 facilities leased from EO-A. Determining the best hubbing strategy requires solving a hub location problem, and the solution specifies which nodes should be designated as hubs, how many multiplexers are required at each hub and how the EO demands should be routed, either directly to the POP (on DS1s) or via a hub.

The two subproblems of the LDAND problem, the dimensioning problem and the hub location problem, require different units of demand. The dimensioning problem deals with traffic demand, expressed in Erlang units, and considers the variation throughout the day. The hub location problem works with the number of circuits leased for DTT at each EO, which is an output of the dimensioning problem. This linkage between the two problems suggests a natural two-step solution approach: determine the number of DS1s to lease between each EO and the POP and then solve the hub location problem. However, this approach ignores the fact that the two aspects of the problem are interdependent. As will be evident, the best hubbing strategy depends, in part, on the number of DS1s leased for DTT from each EO, so the solution of the hub location problem depends on the solution of the dimensioning problem. In addition, because the cost of leasing facilities for DTT depends on the distance between an EO and the hub (or POP) to which it homes, the appropriate number of DS1s to lease, which is the solution of the dimensioning problem, depends on the solution of the hub location problem. Consequently, our model and solution algorithm must consider the interrelationship between the dimensioning problem and the hub location problem.

The rest of this paper is organized as follows. We conclude §1 with a review of the related literature. We present our model for the LDAND problem in §2. In §3, we describe an iterative solution algorithm that combines the queueing aspects of the dimensioning problem with an integer program for the hub location problem. In §4, we present computational results, which indicate that our solution approach is effective in designing low-cost access networks. We provide concluding remarks in §5.

Related Literature. Broadly defined, local access networks connect customers to a switching center that interconnects with a backbone network. In the associated design problem, demands from multiple source nodes (EOs) must be routed to a single sink node (the POP), and link capacities, selected from a set of standard types, must be assigned. Models for local access network planning have grown out of the work in centralized teleprocessing network design of the 1970s (for example, see Boorstyn and Frank 1977 and McGregor and Shen 1977). The survey papers by Gavish (1991) and Balakrishnan et al. (1991) provide a good introduction to the practical issues and the modeling aspects of local access network design. Researchers have studied several variations of the local access network design problem, with the major differences arising from assumptions on the topology of the underlying physical network, the treatment of existing facilities, and the structure of the facility costs. Balakrishnan et al. (1991) develop a general framework based on a layered network representation that encompasses a broad range of the single-period planning models that have been discussed in the literature.

Early work restricted the access network to a tree structure. Only recently have researchers studied access networks with no restrictions on the underlying physical network. Sherali et al. (2000) model an access network hub location problem that includes multiplexer installation costs with a discrete nonlinear function. The authors apply the reformulation-linearization technique to develop two stronger formulations and design a heuristic that combines an exact solution procedure and a Lagrangian dual-based heuristic. They conduct computational experiments on 80 instances generated using simulated and real data. The hybrid solution approach finds an optimal solution for 70 of the 80 problems and solutions within 2.6% of the best known for the remaining 10.

The local access network design problem is a special case of the more general network design problem studied by Salman et al. (2001). The authors propose a branch-and-bound technique called search by objective relaxation (SOR) to solve problems where capacity is installed in discrete quantities and the associated cost function exhibits economies of scale. The algorithm is based on solving relaxations that are obtained by approximating the noncontinuous function by its convex envelope. For randomly generated test problems, the SOR method solves to optimality problems with 10 nodes and is effective in reducing the integrality gap for problems with 20 and 30 nodes.

Frantzeskakis and Luss (1999) describe a capacity expansion model with the additional complexity that the embedded network may be rearranged. That is, demand may be rerouted and existing facilities may be disconnected or used differently at a cost. The authors formulate the problem as a network flow problem and develop a heuristic to solve it. Computational results on eight realistic test networks, ranging in size from 38 to 89 locations, indicate that the heuristic quickly finds solutions within 0% to 7.6% of optimality. The authors find that there is substantial benefit in allowing demand rearrangement; network costs are 7% to 39% higher when no rearrangement is allowed.

Sherali et al. (2000) and Salman et al. (2001) considered the design of new networks, while Frantzeskakis and Luss (1999) studied capacity expansion. For these three papers, and for the general network design literature, an underlying assumption is that demand is expressed in (discrete) units of circuits. In contrast, our assumption is that demand is expressed in Erlang units, which is typical for voice traffic and, hence, our work must consider the interaction between the queueing problem and the facility location problem. We believe that including the interaction between the two components more accurately models the long-distance access network design problem and provides the potential to telecommunications planners for even greater costs savings.

The contributions of this work to the literature are twofold. First, for a competitive industry where it is critical to reduce costs to be profitable, we describe a methodology to significantly lower the access costs paid by a long-distance company. And second, we explicitly consider the interaction between the queueing (or dimensioning) problem and the facility location problem, an approach that, to our knowledge, is novel to the literature.

2. Problem Formulation

In this section, we present the formulation of the LDAND problem. To begin, we describe how we model the fixed alternate routing policy as a queueing system and how we simultaneously consider the dimensioning and the hub location aspects of the problem in our model.

2.1. Preliminaries

Traffic intensity, a common measure of network demand expressed in Erlangs, is defined as the product of the number of calls during a 1-hour period and the average duration of a call. Although traffic intensity varies from hour to hour, networks typically are analyzed in terms of the average activity (traffic intensity) during the busiest hour of the day, called the busy hour. However, because using only the busy hour can result in overdimensioning of the network resources, we incorporate the hour to hour variability in the LDAND problem. To do so, we assume that, for each EO $i \in N$ (where N denotes the set of EOs), we are given a_{it}, the traffic intensity in Erlangs during time t, where $t = 1..T$.[1]

For the dimensioning aspect of the LDAND problem, we can use a simple overflow model to describe the relationship between the number of DS1s leased at an EO and the number of minutes that are transported through the tandem under a fixed alternate routing policy. We assume that customer arrivals (i.e., call arrivals) to an EO i follow a Poisson process and are offered first to a primary server group of size s, which is the group of s DTT circuits leased for EO i. Customers who find all servers busy in the primary group overflow to an infinite server overflow group, which is the group of shared TST circuits. The probability that all s primary servers are busy, or equivalently the proportion of arriving customers who find all s primary servers busy, is given by the *Erlang B formula*

$$B(s, a) = \frac{a^s/s!}{\sum_{k=0}^{s}(a^k/k!)},$$

where a is the offered load in Erlangs.

[1] The day can be broken up into any number of time periods. It is common in the industry to break it up into hours. Equation (1) can be modified easily if the lengths of the time periods are unequal.

For conventional voice traffic, the size of the server group is measured in terms of 64 kbps voice circuits called DS0. In our model, the DTT circuits will be DS1 facilities, so we must adjust the formula. Then, $B(24s_i, a_{it})$ gives the proportion of the offered load a_{it} that overflows the group of s_i DS1s at EO i ($i \in N$) during time period $t = 1..T$. Multiplying $B(24s_i, a_{it})$ by the offered load a_{it} gives the portion of the offered load in Erlangs that overflows to the shared TST circuits. To convert this value to minutes, we multiply by the length (in minutes) of period t, denoted l_t. Here, $l_t = 60$.

To compute the total number of minutes during the day that are transported using TST, we compute and sum the number of minutes that overflow during each hour of the day. Finally, to get the total number of minutes that overflows for the month, we multiply by the number of days in a month, D. Therefore, for EO $i \in N$, the total number of minutes in a month that overflows the DTT group of s_i circuits, denoted m_i, is given by the following equation:

$$m_i = D\sum_{t=1}^{T} l_t a_{it} B(24s_i, a_{it}). \tag{1}$$

Equation (1) describes the nonlinear relationship between the number of DS1s leased for DTT and the number of minutes of traffic that are transported through the tandem switch under a fixed alternate routing scheme. Next, we incorporate this nonlinear relationship characterizing the dimensioning problem into the formulation of the hub location problem.

2.2. Formulation

Given the offered load in Erlangs during each time interval of the day for each EO and all of the component costs for DTT and TST, the objective of the LDAND problem is to determine the number of DS1s to lease for DTT at each EO, which EOs should be designated as hubs, how many multiplexers are required at each hub and how the EO demands should be routed in such a way as to minimize the total cost of access. To manage the complexity of their networks, long-distance companies often limit the number of hubs to which an EO can home (send traffic to). Let G be a graph with node set $\overline{N} = N \cup \mathrm{POP}$. Let $H \subseteq \overline{N}$ denote the set of candidate hub locations. Next, we define the parameters and decision variables.

Parameters

c_{ij} = monthly cost of leasing 1 DS1 between EO i and hub j, $\forall i \in N$, $\forall j \in H$

b_j = monthly cost of leasing 1 DS3 between hub j and the POP, $\forall j \in H$

k_i = cost per minute of TST from EO i to the POP via the tandem, $\forall i \in N$

T = number of time intervals per day

D = number of days per month

a_{it} = offered load in Erlangs at EO i during time interval t, $\forall i \in N$, $\forall t = 1..T$

l_t = length (in minutes) of time period t, $\forall t = 1..T$

e = limit on the number of hubs to which an EO can home.

Decision Variables

s_i = total number of DS1s leased for DTT at EO i, $\forall i \in N$

m_i = monthly minutes of TST from EO i, $\forall i \in N$

x_{ij} = number of DS1s leased between EO i and hub j, $\forall i \in N$, $j \in H$

y_{ij} = 1 if EO i homes to hub j and 0 otherwise, $\forall i \in N$, $\forall j \in H$

u_j = number of DS3s required between hub j and the POP, $\forall j \in H$.

$$\text{Minimize} \quad \sum_{j\in H} b_j u_j + \sum_{i\in N}\sum_{j\in H} c_{ij}x_{ij} + \sum_{i\in N} k_i m_i \tag{2}$$

$$\text{subject to} \quad D\sum_{t=1}^{T} l_t a_{it} B(24s_i, a_{it}) = m_i \quad \forall i \in N \tag{3}$$

$$\sum_{j\in H} x_{ij} = s_i \quad \forall i \in N \tag{4}$$

$$\sum_{i\in N} x_{ij} - 28u_j \le 0 \quad \forall j \in H \tag{5}$$

$$\sum_{j\in H} y_{ij} \le e \quad \forall i \in N \tag{6}$$

$$x_{ij} - s_i y_{ij} \le 0 \quad \forall i \in N, \forall j \in H \tag{7}$$

$$x_{ij} \in Z^+ \quad \forall i \in N, j \in H \tag{8}$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in N, j \in H \tag{9}$$

$$u_j \in Z^+ \quad \forall j \in H \tag{10}$$

$$s_i \in Z^+ \quad \forall i \in N. \tag{11}$$

In this formulation, the objective function (2) seeks to minimize the sum of the DS3 facility costs (including the multiplexer costs), the DS1 facility costs, and the TST usage costs. This represents the amount paid monthly by the long-distance company to the local telephone company for access. Constraints (3), developed in §2.1, model the fixed alternate routing policy and express the relationship between the number of DTT circuits at an EO and the resulting traffic overflow to the tandem switch. Constraints (4) specify that all DS1s leased for DTT must home to a hub location (which may be the POP). Constraints (5) express the multiplexer capacity constraints; each DS3 facility can carry at most 28 DS1 facilities. (Note that the formulation is easily modified to accommodate other multiplexing relationships and technologies.) Constraints (6) limit the number of hubs to which an EO can home to the maximum value e, which typically is set to 2 or 3 by a long-distance company. Constraints (7) link each x_{ij} variable to a y_{ij} variable; for EO i, the number of DS1s homed to hub j must be no more than the total number of DS1s for EO i if hub j is selected and 0 if hub j is not selected. Constraints (8)–(11) restrict the decision variables x_{ij}, u_j, and s_i to nonnegative integer values and y_{ij} to binary values.

2.3. Solving the LDAND Problem

As presented, the formulation of the LDAND problem is a mixed integer program with a set of nonlinear constraints (3). (Constraint (7) also is nonlinear but is easily made linear by replacing s_i by a suitably large constant.) Replacing the m_i variables in the objective function with the left-hand side of constraints (3) yields an equivalent formulation, an integer program with linear constraints and a nonlinear objective function. In this section, we describe how to linearize the objective function, thereby obtaining an integer linear program, which can be solved using standard techniques.

Let s_{MAX_i} be an upper bound on the number of DS1s leased at EO i. Let L_i be the TST cost for EO i when there are no DS1s at EO i. Define s_{ji} as a binary variable that is 1 if j or more DS1s are leased at EO i and 0 otherwise. Thus, s_{ji} represents the jth DS1 leased at EO i. Let k_{ji} represent the decrease in the TST costs due to the addition of the jth DTT DS1 at EO i. In other words, $k_{ji} = k_i D\sum_{t=1}^{T} l_t a_{it}[B(24(j-1), a_{it}) - B(24j, a_{it})]$. Then, the term $L_i - \sum_{j=1}^{s_{\text{MAX}_i}} k_{ji}s_{ji}$ replaces the $k_i m_i$ term. Adding two sets of constraints, (12) and (13), and slightly modifying constraints (7), gives the following integer linear programming formulation.

$$\text{Minimize} \quad \sum_{j\in H} b_j u_j + \sum_{i\in N}\sum_{j\in H} c_{ij}x_{ij} + \sum_{i\in N}\left(L_i - \sum_{j=1}^{s_{\text{MAX}_i}} k_{ji}s_{ji}\right)$$

$$\text{subject to} \quad (4), (5), (6), (8), (9), (10), \text{ and } (11)$$

$$s_{ji} \le s_{j-1,\,i} \quad \text{for } j = 2..s_{\text{MAX}_i}, \forall i \in N \tag{12}$$

$$\sum_{j=1}^{s_{\text{MAX}_i}} s_{ji} = s_i \quad \forall i \in N \tag{13}$$

$$x_{ij} - s_{\text{MAX}_i} y_{ij} \le 0 \quad \forall i \in N, \forall j \in H$$

$$s_{ji} \in \{0, 1\} \quad \text{for } j = 1..s_{\text{MAX}_i}, \forall i \in N.$$

To apply standard integer programming solution techniques, we need to precompute the k_{ji} values for every feasible number j of DTT DS1s at each EO i. Further, we must introduce s_{MAX_i} additional binary variables for each EO i. An upper bound on s_{MAX_i} can be obtained by determining the cost of a DS1 from EO i to its closest permissible hub (including EO i) and calculating the smallest j for which k_{ji} is less than or equal to this cost (see the appendix). Consequently, the size of the formulation grows quite rapidly with the number of EOs and traffic (as s_{MAX_i} increases with traffic); specifically, the integer linear program has an additional $\sum_{i\in N} s_{\text{MAX}_i}$ binary variables and constraints. As a result, solving the problem is extremely difficult. For all but the smallest problems, a state of the art mixed integer programming solver like CPLEX runs out of memory or requires more than 24 hours of CPU time. Given that this problem needs to be solved for each of the 184 LATAs, typically on a quarterly basis, this approach is not computationally viable for long-distance companies. In the next section, we present a computationally viable approach for solving the LDAND problem—a three-phase iterative solution algorithm that provides high-quality solutions.

3. Algorithm

The nonlinear term in the objective function makes solving the LDAND problem difficult, even with the linearization technique. Suppose, however, that we knew the optimal values of the s_i variables for all EOs $i \in N$; then, the nonlinear term would reduce to a constant value, leaving an integer linear programming problem. This observation motivates the development of our iterative solution algorithm.

To begin, we use a greedy algorithm to compute an initial number of DTT DS1s for each EO. The greedy algorithm assumes that each EO homes to the POP and evaluates the tradeoff between the cost of DS1s and the cost of TST. Next, with the DS1 values fixed, we solve an integer programming problem to determine an optimal hubbing strategy. Once we know the hub locations, it may be possible to decrease the total cost by adjusting the number of DS1s at an EO. This is true because (1) the initial value of DS1s was determined assuming the POP as the hub and now the hub(s) to which an EO can be assigned may be closer than the POP and (2) there may be some DS3s that are carrying fewer than 28 DS1s and, hence, could transport additional DS1s for "free." To determine the set of adjustments that will maximize the decrease in the total cost, we formulate and solve a network flow problem. If the network flow problem adjusts any of the DS1 values, we re-solve the hub location problem. The algorithm iterates between solving the hub location problem and the network flow problem until no changes are made. The iteration between the two optimization problems is the key feature of the algorithm that captures the interrelationship between the dimensioning and the hub location aspects of the LDAND problem. In the following sections, we discuss each step in detail.

3.1. Greedy Algorithm to Determine Initial DS1 Values

The first step of the algorithm, which considers each EO individually, determines an initial number of DS1s to lease for DTT at each location. The essential idea of this step is, starting from zero, to increase the number of DS1s connecting an EO and the POP until the cost of adding a DS1 is greater than the corresponding decrease in TST costs.

As in §2, for each EO $i \in N$, we let s_i denote the number of DS1s leased for DTT. (Recall these DS1s form the primary server group). Let $c_{i\text{POP}}$ denote the cost of leasing one DS1 between EO i and the POP, $m_i(s_i)$ denote the total monthly minutes that overflow the primary server group of size s_i DS1s at EO i (and that are carried as TST) and k_i denote the cost per minute of TST from EO i. Then, the algorithm for determining an initial value for the number of DTT DS1s is as follows.

ALGORITHM *Initial Values.*
For each EO $i \in N$,
0. Set s_i to 0 and compute $c_{i\text{POP}}$ and $m_i(s_i)$.
1. Compute $m_i(s_i + 1)$.
2. If $c_{i\text{POP}} < k_i[m_i(s_i) - m_i(s_i + 1)]$, then increment s_i and go to 1. Otherwise, stop with a primary server group size of s_i.
END OF *Initial Values.*

At each iteration, the greedy algorithm examines whether increasing the number of DS1s by one results in a decrease in the TST costs that is larger in value than the cost of a DS1. Initially, the TST costs for EO i are $k_i m_i(0)$. After adding one DS1, the TST costs for EO i are $k_i m_i(1)$, so the decrease is $k_i[m_i(0) - m_i(1)]$. If the cost of a DS1, $c_{i\text{POP}}$, is less than the decrease in the TST costs, the algorithm adds one DS1. The algorithm continues adding DS1s until the decrease in TST costs is smaller than the cost of a DS1. Notice that $c_{i\text{POP}}$ is constant, while $k_i[m_i(s_i) - m_i(s_i + 1)]$ may change during the course of the algorithm.

To establish the correctness of the greedy algorithm, we must show that the change in TST costs is monotonically decreasing. In showing this result, we use the convexity property of the Erlang B formula (see Messerli 1972). The property, which is a commonly accepted fact that is basic to the process of economic alternate routing in traffic engineering, states that, for a server group with sequential assignment of offered calls, the load carried on the last server is monotonically decreasing with the number of trunks.

Theorem 1 (Messerli 1972). *The sequence $a_1, a_2, \ldots,$ where $a_i = a[B(i-1, a) - B(i, a)]$ satisfies $a_1 > a_2 > \cdots$ for any positive load a.*

Because we are working with DS1s, we need a modified version of this result. From Theorem 1, it follows that $B(t-1,a)-B(t,a) > B(u-1,a)-B(u,a)$ for any $t < u$. Therefore, for any k,

$$\sum_{j=t}^{t+k}(B(j-1,a)-B(j,a)) > \sum_{j=u}^{u+k}(B(j-1,a)-B(j,a)) \quad \text{for } t < u.$$

Cancelling terms and substituting $t=\alpha(i-1)+1$, $u=\alpha i+1$, and $k=\alpha-1$, we obtain

$$B(\alpha(i-1),a)-B(\alpha i,a) > B(\alpha i,a)-B(\alpha(i+1),a) \quad \text{for } \alpha > 0 \text{ and integer.} \tag{14}$$

For $\alpha = 24$, Equation (14) shows that, when traffic is offered sequentially to the server group, the load carried on the ith DS1 is monotonically decreasing. From Equations (1) and (14), it follows that $m_i(s_i) - m_i(s_i+1)$ is monotonically decreasing, which is the desired result.

Since the cost of leasing a DS1 is a function of distance (see Table 1), the value of $c_{i\text{POP}}$ is a function of the distance between EO i and the POP. Hence, from Step 2 of *Initial Values*, the number of DS1s to lease at EO i depends on the distance from EO i to the POP. It follows that, for two EOs with identical traffic patterns, the algorithm will assign a greater or equal number of DS1s to the EO that is closer to the POP than to the EO that is further away.

Running Time of *Initial Values*. Each pass of the algorithm requires $O(1)$ time. For each EO $i \in N$, the algorithm performs $s_i + 1$ passes through the algorithm, where s_i is the number of DTT DS1s that prove in. Therefore, the total number of steps is $O(\sum_{i\in N} s_i + |N|)$. The running time is linear in the number of DS1s that are added, so the algorithm is pseudopolynomial.

3.2. Integer Programming Model for the Hub Location Problem

Given a value for the number of DS1s leased at each EO, the second step of the algorithm determines a hubbing strategy by solving a hub location problem. The objective is to determine which EOs should be designated as hubs, how many multiplexers are required at each hub, and how the DS1s connecting each EO and the POP should be routed in such a way as to minimize the total cost of DS1 facilities and DS3 facilities (including multiplexers).

3.2.1. Initial Formulation. In the hub location problem, the s_i values are known, so the corresponding m_i values can be computed. Therefore, the nonlinear term of the objective function ($\sum_{i\in N} k_i m_i$) of the integer program presented in §2.2 becomes a constant and can be removed from the formulation. Hence, we can write the formulation of the hub location problem as follows:

$$\text{(HLP)} \quad \text{Minimize} \quad \sum_{j\in H} b_j u_j + \sum_{i\in N}\sum_{j\in H} c_{ij} x_{ij}$$
$$\text{subject to} \quad (4), (5), (6), (7), (8), (9), \text{ and } (10).$$

Our formulation HLP bears some resemblance to the LAN formulation of Sherali et al. (2000) for the LATA network design problem. There are a number of differences, however. First, Sherali et al. (2000) view the POP node as a facility with unlimited capacity, whereas we view the POP node as a candidate hub and as such require the model to determine its multiplexer requirements. (The cost of the multiplexers required at the POP is included in the objective function.) Second, the two formulations handle the hubbing restrictions entirely differently. Sherali et al. (2000) do not limit the number of hubs to which an EO can home but do impose a lower and an upper bound on the total number of multiplexers allowed. Their objective function then includes a cost for installing the facilities, which is a function of the total number of facilities. In the HLP formulation, motivated by our industry experience, we limit the number of hubs to which an EO can home. An additional restriction on the total number of facilities allowed and a corresponding term in the objective function easily can be included but are not necessary in our application.

3.2.2. Analysis of HLP. In this section, we study the underlying structure of the HLP formulation and identify several characteristics of the model that impact the solution algorithm.

Strengthening the LP Relaxation. As formulated, the objective function value of the LP relaxation provides a weak bound on the optimal (integer) objective function value of HLP. For the set of 70 test problems (which will be described in §4), the value of the LP relaxation is in the range of 34.9% to 75% of the optimal integer value and averages 57.4%. To strengthen the LP relaxation of HLP, we add the following set of constraints:

$$y_{ij} - u_j \le 0, \quad \forall i \in N, \forall j \in H. \tag{15}$$

These constraints specify that if DS1s from EO i home to hub j, then there must be at least one DS3 connecting hub j and the POP. Adding these constraints dramatically improves the bound provided by the LP relaxation. For the same set of test problems, the value of the LP relaxation is in the range of 98.4% to 100% of the optimal integer value and averages 99.7%.

Relaxing the Integrality on the x_{ij} Variables. Since x_{ij} represents the number of DS1s from EO i that home to hub j, we restrict the x_{ij} variables to nonnegative integer values in HLP. Although we allow a group of DS1s to be split among multiple hubs, each individual DS1 must be considered as a discrete indivisible unit. Due to the structure of HLP, however, we can relax the integrality restriction on the x_{ij} variables and still obtain integer-valued solutions. To prove the claim, we show that the x_{ij} constraint matrix is totally unimodular.

Consider HLP. If $y_{ij} \in \{0, 1\}$ and $u_j \in Z^+$, then let $\mathbf{A}$ denote the x_{ij} constraint matrix comprised of the following rows:

$$\sum_{j\in H} x_{ij} = s_i \quad \forall i \in N \tag{4}$$

$$\sum_{i\in N} x_{ij} \le 28u_j \quad \forall j \in H \tag{5}$$

$$x_{ij} \le s_i y_{ij} \quad \forall i \in N,\ \forall j \in H. \tag{7}$$

By Proposition 2.1 of Nemhauser and Wolsey (1999, p. 540), we can ignore the unit rows of the matrix, i.e., constraints (7). Observe that, in the remaining matrix, each column has exactly two nonzero elements. Let Q_1 and Q_2 denote a partition of $\mathbf{A}$ in which Q_1 contains the rows corresponding to constraints (4) and Q_2 contains the rows corresponding to the constraints (5). Then, by Corollary 2.8 (Nemhauser and Wolsey 1999, p. 544), the matrix is totally unimodular. Since the x_{ij} constraint matrix is totally unimodular, the extreme points of the linear program $\mathbf{Ax} \le \mathbf{b}$ are integral for all integral right-hand side (RHS) vectors $\mathbf{b}$. The RHS of (4), (5), and (7) are integer when $y_{ij} \in \{0, 1\}$ and $u_j \in Z^+$, which gives the desired result.

An interesting consequence of the total unimodularity of the x_{ij} constraint matrix is that we can develop an equivalent formulation (as in Sherali et al. 2000) in which we let X_{ij} denote the fraction of the DS1s from EO i that home to hub j. Then, we can write the following formulation:

$$\begin{aligned}
(\overline{\text{HLP}})\quad \text{Minimize}\quad & \sum_{j\in H} b_j u_j + \sum_{i\in N}\sum_{j\in H} c_{ij} s_i X_{ij} \\
\text{subject to}\quad & (6),\ (9),\ \text{and}\ (10) \\
& \sum_{j\in H} X_{ij} = 1 \quad \forall i \in N \\
& \sum_{i\in N} s_i X_{ij} - 28u_j \le 0 \quad \forall j \in H \\
& X_{ij} - y_{ij} \le 0 \quad \forall i \in N,\ \forall j \in H \\
& X_{ij} \ge 0 \quad \forall i \in N,\ j \in H.
\end{aligned}$$

Notice that we can obtain this formulation by setting $X_{ij} = x_{ij}/s_i$ and substituting for x_{ij} in HLP. In terms of the continuous variables, x_{ij}, these two formulations are equivalent, because we have shown that the x_{ij} variables in HLP do not need to be restricted to integers.

Comparison to Capacitated Facility Location Problem (CFLP). In its equivalent form shown above, the resemblance between our facility location problem and the well-known CFLP is more noticeable. There are two major differences, however. First, there are no hubbing restrictions in the CFLP; demand can be split among any number of locations. In HLP, demand can be split, but there are restrictions on the number of hubs. (The hubbing restrictions require the introduction of the y_{ij} variables, which are not necessary in CFLP.) Second, in CFLP, the facility location variable (u_j) is binary; at any location, at most, one "bundle" of capacity is available. In HLP, the variable u_j is integer; at any location, multiple bundles of capacity may be available.

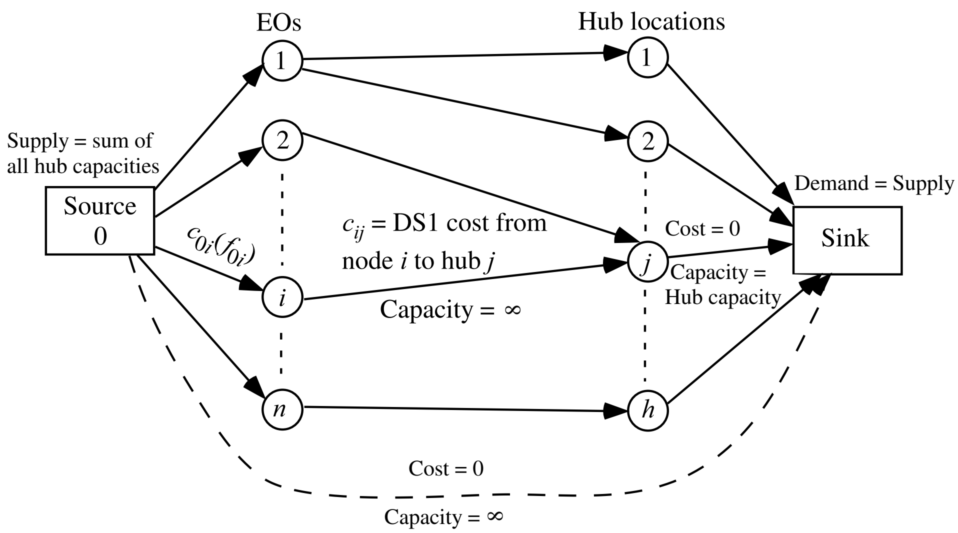
These two differences affect the form of the constraints used to strengthen the LP relaxation. The strong formulation of CFLP includes the set of facet defining constraints $X_{ij} \le u_j$, $\forall i \in N$, $\forall j \in H$, whereas HLP (or $\overline{\text{HLP}}$) includes the set of constraints $y_{ij} \le u_j$, $\forall i \in N$, $\forall j \in H$. Both specify that there must be at least one "facility" at location j when demand from location i is assigned to j. For HLP (or $\overline{\text{HLP}}$), $y_{ij} \le u_j$ is preferred since that automatically specifies the condition $x_{ij} \le s_i u_j$ ($X_{ij} \le u_j$ in $\overline{\text{HLP}}$) and adds an additional restriction. The running times of HLP and $\overline{\text{HLP}}$ are comparable, so either formulation can be used. In our computational experiments, we use HLP because specifying the *number* of DS1s that home to a particular hub is more natural in the context of our application than specifying the *fraction* of DS1s.

3.3. Modifying the DS1 Values: Packing Algorithm

At the completion of the first two steps, the algorithm has determined initial values for the number of DS1s to lease at each EO and has specified a hubbing strategy. In the solution of the hub location problem, there may be a DS3 leased between a hub and the POP that is not fully utilized (i.e., packed with fewer than 28 DS1s). If so, it may be possible to decrease the overall cost by increasing the number of DTT DS1s from one or more of the EOs that can send traffic to that hub, because, in effect, we already have paid for the DS3.

To determine the best adjustments to the DS1 values that reduce the overall access costs, we would like to solve the following problem. Given the hub locations, the number of DS3 facilities at each hub location and any restrictions on hubbing requirements for each EO, determine the number and assignment of DS1s for each EO to minimize the sum of the DTT and TST costs. We now describe how to model this problem

Figure 2 Modeling the Packing Problem as a Convex Cost Flow Problem on a Bipartite Graph



as a convex cost network flow problem on a bipartite graph. First, we ignore the constraint limiting an EO to be assigned to e or fewer hub nodes.

3.3.1. Network Flow Problem. To construct the bipartite graph as shown in Figure 2, we begin by creating two sets of nodes. Set $N = \{1, 2, \ldots, n\}$ contains one node for each EO, and set $H = \{1, 2, \ldots, h\}$ contains one node for each EO at which we have located a hub. We create a set of arcs from the EOs (N) to hubs (H), creating an arc from EO i to hub location j if connecting EO i to hub location j is permitted. The cost of this arc is denoted by c_{ij} and is set equal to the cost of leasing a DS1 between EO i and hub location j. The capacity of this arc is ∞. Next, we create a sink node and connect each hub location node to the sink node with a directed arc from the hub location to the sink node. For each arc from a hub location j to the sink node, the cost of the arc is 0, and the capacity of the arc is equal to the capacity of the hub (i.e., the number of DS1s of traffic it can accommodate). Finally, we create a source node and connect it to each EO location i. For each arc from the source node to an EO i, the cost of the arc is denoted by $c_{0i}(f_{0i})$ and is equal to the change in the TST costs that results when f_{0i} DS1s are leased for DTT at EO i. In other words, $c_{0i}(f_{0i}) = k_i[m_i(f_{0i}) - m_i(0)]$. (The flow requirements for this cost function are integer.)

Observe that the cost function $c_{0i}(f_{0i})$ takes on negative values and is a (piecewise linear) convex cost function if we apply linear interpolation between two points within the integer flow bounds. Therefore, finding an optimal solution to the packing problem is equivalent to solving a convex cost flow problem with a piecewise linear convex cost function.

As described in Ahuja et al. (1993, pp. 551–556), we can transform this problem into a minimum cost flow problem and solve it using a successive shortest path algorithm in the absence of negative cost cycles. The bipartite network in Figure 2 is acyclic (and, thus, contains no negative cost cycle), so the successive shortest paths algorithm may be applied.

The successive shortest path algorithm for minimum cost flows (see Ahuja et al. 1993, p. 320) finds a minimum cost path from a source node (supply node) to a sink node (demand node) and sends as much flow as possible along the path. After augmenting the flow, the algorithm updates the *residual network*.[2] This process of finding a shortest path, augmenting flow, and updating the residual network continues until all of the flow has been sent from the source nodes to the sink nodes.

Next, we describe how the successive shortest path algorithm can be applied in the context of solving the packing problem. To transform the convex cost flow problem into a minimum cost flow problem, we expand the underlying network by replacing each single arc connecting the source node to EO i, denoted $(0, i)$, with multiple copies of the arc, $(0, i)^1, (0, i)^2, (0, i)^3, \ldots$. Each arc $(0, i)^l$ has a capacity of one unit and a cost equal to $k_i[m_i(l) - m_i(l-1)]$. Since the capacity of each arc $(0, i)^l$ is one, the successive shortest path algorithm will send one unit of flow on the minimum cost path from the source to the sink. This observation permits us to maintain a single copy of the arc $(0, i)$ in the network instead of multiple copies. As the algorithm progresses, the cost of the arc is changed to reflect the cost of the appropriate copy

[2] Recall the residual network is created by replacing an arc (i, j) carrying f_{ij} units of flow with capacity u_{ij} and cost c_{ij} by two arcs—a forward arc (i, j) with capacity $u_{ij} - f_{ij}$ and cost c_{ij} and a reverse arc (j, i) with capacity f_{ij} and cost $-c_{ij}$.

of arc $(0, i)$ that should be used in any flow augmentation step.

Because there are no negative cost cycles in the network, the shortest path from the source to the sink in the residual network will not include reverse arcs into the source node or reverse arcs out of the sink node. Consequently, we need not construct these arcs in the residual network. Additionally, because each flow augmentation is one unit, we do not need to keep track of the capacity of the reverse arcs from the hub location nodes to the EO nodes in the residual network; if the reverse arc is in the residual network, it will have a capacity of at least one. Finally, when the algorithm determines that the shortest path from the source to the sink contains only the arc from the source to the sink with a cost of zero, the implication is that no additional savings can be obtained by increasing the number of DS1s at an EO. Therefore, we can stop the algorithm and conclude that we have solved the packing problem for the DS1 values and the hub assignments.

The following algorithm more formally describes how to solve the packing problem.

ALGORITHM *Packing.*

0. Set all flow variables to 0.

1. Set costs of arcs from source node to EO i equal to $c_{0i}(f_{0i}+1) - c_{0i}(f_{0i})$. (This represents the change in TST cost for additional DS1s, which is negative.)

2. Find shortest path from source node to sink node in the *residual graph* (use cost defined in step 1 for cost on arcs $(0, i)$).

3. If cost of shortest path is negative, augment one unit of flow on shortest path.

4. Update residual network and costs of arcs defined in step 1. The residual graph is obtained by creating a reverse arc (j, i) for every forward arc (i, j) with positive flow. The cost of the reverse arc (j, i) is $-c_{ij}$ and its capacity is equal to the flow on arc (i, j).

5. Repeat Steps 1–4 until we find a shortest path from source to sink with nonnegative cost (indicates no more savings associated with moving TST to DTT with current hubbing arrangement).

END OF *Packing.*

At the conclusion of *Packing*, the flow variables f_{0i} specify the total number of DS1s to lease at EO i, while flow variables f_{ij} specify the number of DS1s from EO i assigned to hub location j.

Running Time of *Packing*. The number of augmentations is bounded by the total hub capacity, which we denote by TOTHUBCAP. Each augmentation requires finding a shortest path on a bipartite graph, which takes $O(nh + (n + h)\log(n + h))$ time. Therefore, the total running time is $O(\text{TOTHUBCAP} \cdot (nh + (n + h)\log(n + h)))$, which is pseudopolynomial.

Before we discuss how to deal with the hubbing restrictions, we observe that Step 1, *Initial Values*, also may be interpreted as a special case of *Packing*. In this case, the POP is the only hub with unlimited capacity.

3.4. Hubbing Restrictions and Iterative Algorithm for LDAND Problem

In this section, we describe how the packing algorithm incorporates the constraint that an EO be assigned to at most e hubs. In the network flow problem, this constraint corresponds to ensuring that there are no more than e arcs out of an EO with positive flow. This may be viewed as a "degree" constraint on the EO nodes N in the bipartite graph. In the case that $e = 1$ or $e = 3$, we can show that the minimum cost flow problem with degree constraints on nodes is $\mathcal{NP}$-hard (see appendix). We strongly suspect the problem is $\mathcal{NP}$-hard when e is fixed; consequently, we develop two heuristics to deal with this constraint. We describe these heuristics, H1 and H2, in conjunction with the iterative algorithm for LDAND problem.

Heuristic 1 (H1). Our first heuristic, called H1, combines the procedures described in the three previous sections in an iterative algorithm as follows. It begins with Step 1 the greedy algorithm *Initial Values*, to determine an initial value for the number of DS1s at each EO. Next, the algorithm solves Step 2, the hub location problem, to find an optimal hubbing strategy. Note, the hub location problem includes only EOs with DS1 values strictly greater than zero. For Step 3, *Packing*, the bipartite graph is constructed according to the solution of Step 2: an arc is created from EO i to hub location j only if $y_{ij} = 1$, which automatically ensures that the degree constraints are satisfied. If packing the DS3 facilities reduces the cost of the solution, the algorithm returns to Step 2, re-solves the hub location problem with the new demands and follows to Step 3 as described above. Otherwise, the algorithm stops with a solution for the LDAND problem.

Heuristic 2 (H2). Our second heuristic, called H2, applies Steps 1 and 2 as described above and iterates between the hub location problem and network flow problem just as H1. However, Step 3 is modified to consider sequentially three variants of the network flow problem.

Step 3a (Unconstrained Packing). Create the bipartite network without restricting the connections between EOs and hubs—the network includes an arc from each EO to each hub. Apply *Packing*. If the solution satisfies the degree constraints, it is optimal for the network flow problem with the degree restrictions. Report the solution and exit. Otherwise, go to Step 3b.

Step 3b (Heuristic Packing). Create the bipartite network as in Step 3a, but apply a *modified* packing algorithm, as a heuristic, to obtain solutions that satisfy the degree requirements. If the solution does not increase the access costs, report it and exit. Otherwise, go to Step 3c. The *modified* packing algorithm attempts to find the shortest path from source to sink that does not cause a violation of the degree constraints when flow is augmented along the path. If an EO node has exactly e arcs with positive flow out of it, we do not consider any of the (forward) arcs with zero flow out of that node while searching for the shortest path from the source to the sink. With this change, it is possible for the residual network to have a negative cost cycle (indicating that the cost of the solution can be improved by sending a unit of flow around the cycle, i.e., reassigning some of the DS1s). Consequently, we use a label correcting algorithm for the shortest path problem (modified as discussed above) to find a path whose augmentation does not cause a violation of the degree constraint or to detect and identify a negative cost cycle if present. If a negative cost cycle is found, we augment one unit of flow around it, and if a negative cost path is found, we augment flow on it.

Step 3c (Restricted Packing). Create the bipartite network according to the solution of Step 2 as in H1. Apply *Packing* and report the solution.

In H1, observe that any EO with no DTT DS1s after *Initial Values* will have no DTT DS1s for the entire algorithm. Therefore, H1 cannot accommodate the possibility that once hubs are located at nodes other than the POP, DS1s may become economical for EOs that had no DTT in Step 1. H2 accommodates this possibility and considers all EOs as candidates for DTT in Step 3. Consequently, we expect the quality of solutions produced by H2 to be better than H1. Both algorithms H1 and H2 converge finitely, as the total cost of the solution (DTT costs + TST costs) monotonically decreases in each iteration.

4. Computational Results

In this section, we report on the computational experiments conducted to test the effectiveness of our algorithms. The implementation uses the CPLEX 7.1 Callable Library to solve the hub location problem in Step 2. All experiments were conducted on a Sun Blade 1,000 workstation with two 750-MHz Ultra SPARC III processors and 1 GB RAM.

In presenting our results, we compare the cost of the solution produced by our algorithm with the costs of solutions representative of industry practice. According to our industry contacts, the complexity of an access network design varies from company to company. Smaller long-distance companies tend to rely only on TST, whereas larger long-distance companies tend to use a combination of DTT and TST. Typically, they might determine the number of facilities to lease for DTT with a procedure similar to Step 1 of our algorithm. A company with an operations research group might recognize that DTT costs can be reduced by hubbing and thereby use an existing facility location model to solve for hub locations and EO assignments. As far as we know, however, our iterative method, which considers the interrelationship between the queueing problem and the facility location problem, represents a new approach both in the literature and in practice. Therefore, we will compare our solution with the following alternate solutions: (1) all TST, (2) a combination of DTT and TST but no hubs (Step 1 only), and (3) a combination of DTT and TST with hubs (Step 1 followed by Step 2).

4.1. Data

Recall from the description of the LDAND problem in §2.2, the following data elements within a LATA are required: the locations of all of the EOs, the POP and the tandem switch, the offered load in Erlangs during each time interval for each EO, and all of the component costs for DTT and TST. In this section, we describe the details of each data element in turn.

Network Data. For our study, we selected LATAs encompassing 10 medium and large cities in the United States. For each LATA, we obtained the locations of the EOs and the tandem switch from the Local Exchange Routing Guide, which is a database of local exchange network configuration information maintained by Telcordia Technologies. And, as is common in the industry, we assumed that the POP and the tandem switch are colocated.

Offered Load. The values for the offered load during each time interval of the day for each EO in a LATA are computed from the total volume of long-distance traffic for the LATA, a LATA-to-EO distribution and an EO traffic distribution profile.

For each LATA, the total number of long-distance calling minutes is computed as the product of the population of the LATA (obtained from 1990 U.S. Census data) and an average value of 165 minutes of long-distance calls per person per month. Then, to determine the number of minutes carried by our hypothetical long-distance company, we multiply the total minutes for the LATA by the company's market share. To distribute our company's share of the total minutes for the LATA to the individual EOs, we use a LATA-to-EO distribution provided by a telecommunications company. Such a distribution also could be obtained by mapping the Census data to the EOs.

A traffic distribution profile, which describes how the traffic intensity varies throughout the day, specifies the fraction of the total daily traffic to assign to each time interval of the day. Because business traffic and residential traffic produce different network loading patterns, we need a separate profile for each. To obtain representative profiles, we analyzed the long-distance traffic on the University of Maryland telephone network during the month of September 1999. At the university, one switch handles business traffic generated by faculty, staff, and administrative offices, whereas a second switch handles residential traffic generated by student residences. Therefore, for each traffic type, we were able to create a separate distribution of average traffic intensity by hour of the day and by day of the week. Then, using a weighted average, we created one distribution for each hour of the day for each traffic type. Finally, we normalized the values to obtain the desired fraction of the total traffic assigned to each time interval for each traffic type. Although here we assume that each EO has the same traffic profile, the algorithm can handle individual EO profiles.

Costs. The lease costs for the DTT facilities and the usage charges for TST are available in the tariff documents filed with the FCC. Each local exchange carrier has different charges, so in practice, we would use the appropriate costs for each LATA. For the computational results reported here, however, we use the same costs for all of the test problems. For each of the cost components listed in Table 1, the value represents an average of the costs of four regional bell operating companies, as reported in November 1999 tariffs. All of the costs are expressed as monthly recurring charges.

4.2. Test Problems

A number of factors are likely to affect the performance of our algorithm. In particular, we expect the effectiveness of the algorithm will vary with the volume of demand (or market share), the mix of residential and business traffic, and the relative costs of DS1s, DS3s, and TST charges. We have designed our experiments to evaluate the impact of these factors.

To begin, we define a base test problem for each LATA in which we assume that the market share is 20%, the traffic mix is 50% residential and 50% business, and the costs are as they appear in Table 1. The number of EOs and the total monthly minutes of use (MOU) for each base instance are shown in Table 2. For each LATA, we create six additional instances by testing two alternate values for each of the three factors individually. Table 3 shows the factor values for each instance. In the table, "Base" refers to the original costs shown in Table 1.

4.3. Results

Tables 4, 5, and 6 summarize the performance of H1 and H2 on the 70 test problems. The costs reported represent the network expense costs paid by the long-distance company to the local exchange carrier each month for access. The results indicate that our algorithms produce cost effective access network designs and, as expected, H2 produces lower cost designs than H1. The H2 (H1) solutions are on average 40.4% (41.9%) and 73.3% (76.1%) of the corresponding costs for a network of TST only (denoted TST) and a network without hubs (denoted NH). As compared to the solution obtained by completing Step 1 followed by Step 2 (denoted Iter 1), H2 (H1) results in savings of 0.5% (0.3%) to 21.4% (20.4%) with an average of 8.0% (4.2%).

H2 produces lower cost designs than H1 because, throughout the algorithm, it considers adding DS1s to EOs for which DS1s were not economical in *Initial Values*. The impacts of increasing the number of EOs with strictly positive DS1 values are (1) that the possible number of DS1 adjustments increases, which sometimes increases the number of iterations, and (2) that the size of the integer program (hub location problem) increases, which typically increases the running time. For most instances, H2 requires more iterations than H1, but the number remains reasonable. Fifty-one completed in four or fewer iterations, and 18 completed in five or six. Just one instance, the largest, required eight. For 56 of the 70 instances, H2 requires more CPU time than H1, but the increase is unpredictable. Overall, the CPU times for H2 are quite reasonable. Fifty-six (80%) instances solved in less than one hour with another 10 solving in one to eight hours and two in 11 to 13 hours. The two

Table 2 Number of EOs and Total MOU for Base Problems

ID	# EO	Total MOU
1	75	59,438,808
2	67	71,028,012
3	99	84,977,211
4	87	88,722,315
5	86	92,430,228
6	90	137,505,390
7	132	151,611,966
8	110	171,723,915
9	149	214,492,080
10	163	248,803,830

Table 3 Parameter Values for Test Problem Instances

Mkt share (%)	Mix (R/B) (%)	Costs
20	50/50	Base
20	80/20	Base
20	20/80	Base
10	50/50	Base
40	50/50	Base
20	50/50	$DS3 = 7 * DS1$
20	50/50	$1.2 * TST$

Table 4 Summary of Computational Results: Market Share

					H1			H2							
ID	Mkt (%)	TST ($)	NH ($)	Iter 1 ($)	Cost ($)	Time(s)	Iter	Cost ($)	Time(s)	Iter	H2/TST	H2/NH	H2/I1	H2/H1	¢/min
1	10	89,253	59,966	59,757	59,506	1	2	59,432	1	2	0.666	0.991	0.995	0.999	0.200
	20	178,506	115,789	106,594	88,948	1	2	88,634	2	3	0.497	0.765	0.832	0.996	0.149
	40	357,012	221,921	181,134	150,581	5	2	149,798	9	2	0.420	0.675	0.827	0.995	0.126
2	10	75,763	36,295	33,046	32,099	62	2	31,832	47	2	0.420	0.877	0.963	0.992	0.090
	20	151,527	65,377	55,831	54,295	61	2	52,969	65	3	0.350	0.810	0.949	0.976	0.075
	40	303,053	123,450	90,507	87,009	179	3	84,052	229	3	0.277	0.681	0.929	0.966	0.059
3	10	95,177	50,959	45,582	45,398	1	2	45,119	22	2	0.474	0.885	0.990	0.994	0.106
	20	190,353	93,068	77,731	76,625	177	2	75,736	233	2	0.398	0.814	0.974	0.988	0.089
	40	380,707	174,407	128,832	126,596	803	3	123,732	694	3	0.325	0.709	0.960	0.977	0.073
4	10	122,256	79,132	69,486	68,068	1	2	67,415	31	2	0.551	0.852	0.970	0.990	0.152
	20	244,513	149,074	123,502	118,516	84	3	115,879	49	4	0.474	0.777	0.938	0.978	0.131
	40	489,025	286,809	210,010	174,402	42	3	166,714	120	4	0.341	0.581	0.794	0.956	0.094
5	10	117,691	74,767	58,989	55,428	2	2	53,268	2	2	0.453	0.712	0.903	0.961	0.115
	20	235,383	136,126	93,280	89,330	327	2	85,467	870	3	0.363	0.628	0.916	0.957	0.092
	40	470,765	259,245	155,157	141,634	3,214	2	133,542	1,938	3	0.284	0.515	0.861	0.943	0.072
6	10	158,688	80,674	72,960	71,238	91	3	70,885	84	3	0.447	0.879	0.972	0.995	0.103
	20	317,376	153,590	125,934	121,193	29	3	118,397	147	3	0.373	0.771	0.940	0.977	0.086
	40	634,752	293,582	209,519	193,457	31	3	185,389	236	5	0.292	0.631	0.885	0.958	0.067
7	10	175,319	94,798	78,577	78,040	321	2	76,554	543	3	0.437	0.808	0.974	0.981	0.101
	20	350,638	173,805	134,120	131,458	291	2	127,249	2,585	3	0.363	0.732	0.949	0.968	0.084
	40	701,276	331,648	228,537	221,473	289	2	210,373	2,231	5	0.300	0.634	0.921	0.950	0.069
8	10	200,754	107,377	87,404	82,064	243	3	77,232	257	2	0.385	0.719	0.884	0.941	0.090
	20	401,508	197,789	139,715	129,986	3,026	3	123,663	4,894	4	0.308	0.625	0.885	0.951	0.072
	40	803,016	377,939	238,932	231,807	1,156	3	209,911	5,302	6	0.261	0.555	0.879	0.906	0.061
9	10	277,383	163,804	125,583	120,939	661	3	114,046	1,570	5	0.411	0.696	0.908	0.943	0.106
	20	554,765	310,294	212,256	197,377	1,086	2	189,623	3,021	3	0.342	0.611	0.893	0.961	0.088
	40	1,109,530	603,191	367,705	329,331	49,164	3	305,187	129,309	5	0.275	0.506	0.830	0.927	0.071
10	10	299,905	165,947	119,746	118,570	693	2	107,690	1,146	5	0.359	0.649	0.899	0.908	0.087
	20	599,810	312,670	197,448	192,731	2,685	2	180,394	28,936	5	0.301	0.577	0.914	0.936	0.073
	40	1,199,621	602,336	341,499	333,799	40,774	4	310,642	143,756	3	0.259	0.516	0.910	0.931	0.062

largest instances, Problems 9 and 10 with a 40% market share, took more than 35 hours to solve. It is important to note that the CPU times are dominated by the time spent to solve Step 2—the hub location problem. (Steps 1 and 3 are almost instantaneous.) And, often, within Step 2, good solutions are found early with the majority of the time spent proving optimality. As a result, should the longer running times be impractical in an industrial setting, the solution of the hub location problem could be terminated early instead of being solved to optimality and/or the algorithm could be terminated early by truncating the number of iterations. Throughout the remainder of this section, we will restrict our comments to the performance of H2, since it outperforms H1. H2 produces cost effective access network designs and substantially improves upon techniques representative of industry practice.

Market Share. In the base test problems, we assume that our long-distance company captures 20% of the total calling minutes for the LATA. To evaluate the impact of traffic volume on the results, we also consider market shares of 10% and 40%, which are equivalent to multiplying the total minutes for the LATA by 0.5 and by 2.0, respectively (see Table 4 for the results). As expected, the cost per minute (¢/min) decreases with increasing volume, because DTT proves to be economical for more EOs. As the volume increases and, hence, the number of DS1s increases, the savings achieved by implementing a hubbing strategy increase. For each base problem, the cost of H2 relative to the cost of NH (H2/NH) substantially decreases as volume increases. The gains achieved by H2 relative to Iter 1 (H2/I1) also increase with volume. The solution times increase with volume; increasing the volume tends to increase the number of EOs with DS1s, which increases the size of the hub location problem. The number of hub locations selected increases with volume as well. In fact, the number of hub locations increases almost in direct proportion to the volume increase (see Table 7).

Traffic Mix. Because residential and business traffic are distributed differently throughout the day, we expect the mix of traffic to affect the results. Business

Table 5 Summary of Computational Results: Traffic Mix

					H1			H2							
ID	Mix	TST ($)	NH ($)	Iter 1 ($)	Cost ($)	Time(s)	Iter	Cost ($)	Time(s)	Iter	H2/TST	H2/NH	H2/I1	H2/H1	¢/min
1	80/20	168,944	123,710	120,339	101,744	3	3	101,269	6	3	0.599	0.819	0.842	0.995	0.170
	50/50	178,506	115,789	106,594	88,948	1	2	88,634	2	3	0.497	0.765	0.832	0.996	0.149
	20/80	188,068	129,243	122,909	97,854	1	2	96,616	5	4	0.514	0.748	0.786	0.987	0.163
2	80/20	143,410	83,853	76,534	72,723	142	3	71,900	65	3	0.501	0.857	0.939	0.989	0.101
	50/50	151,527	65,377	55,831	54,295	61	2	52,969	65	3	0.350	0.810	0.949	0.976	0.075
	20/80	159,643	75,760	61,410	60,441	115	2	60,135	266	3	0.377	0.794	0.979	0.995	0.085
3	80/20	180,157	114,177	104,762	101,320	251	2	99,373	122	3	0.552	0.870	0.949	0.981	0.117
	50/50	190,353	93,068	77,731	76,625	177	2	75,736	233	2	0.398	0.814	0.974	0.988	0.089
	20/80	200,550	107,291	86,017	84,326	74	2	82,711	237	3	0.412	0.771	0.962	0.981	0.097
4	80/20	231,415	166,634	150,956	148,620	53	2	140,828	102	4	0.609	0.845	0.933	0.948	0.159
	50/50	244,513	149,074	123,502	118,516	84	3	115,879	49	4	0.474	0.777	0.938	0.978	0.131
	20/80	257,610	170,765	137,278	133,319	5	3	124,582	153	5	0.484	0.730	0.908	0.934	0.140
5	80/20	222,774	156,156	133,738	127,187	196	3	121,913	523	5	0.547	0.781	0.912	0.959	0.132
	50/50	235,383	136,126	93,280	89,330	327	2	85,467	870	3	0.363	0.628	0.916	0.957	0.092
	20/80	247,991	160,306	110,432	103,410	259	3	91,326	941	5	0.368	0.570	0.827	0.883	0.099
6	80/20	300,376	186,260	172,251	166,824	178	3	162,615	96	3	0.541	0.873	0.944	0.975	0.118
	50/50	317,376	153,590	125,934	121,193	29	3	118,397	147	3	0.373	0.771	0.940	0.977	0.086
	20/80	334,376	178,269	143,571	138,510	67	3	131,170	244	4	0.392	0.736	0.914	0.947	0.095
7	80/20	331,856	209,664	186,917	183,135	161	2	176,015	3,139	6	0.530	0.840	0.942	0.961	0.116
	50/50	350,638	173,805	134,120	131,458	291	2	127,249	2,585	3	0.363	0.732	0.949	0.968	0.084
	20/80	369,420	201,662	151,034	149,321	599	2	139,853	2,091	5	0.379	0.694	0.926	0.937	0.092
8	80/20	380,001	233,945	202,741	193,675	130	3	186,220	7,469	4	0.490	0.796	0.919	0.962	0.108
	50/50	401,508	197,789	139,715	129,986	3,026	3	123,663	4,894	4	0.308	0.625	0.885	0.951	0.072
	20/80	423,015	230,177	165,920	162,919	455	2	139,269	14,200	5	0.329	0.605	0.839	0.855	0.081
9	80/20	525,049	355,762	301,195	292,509	477	4	277,954	3,019	6	0.529	0.781	0.923	0.950	0.130
	50/50	554,765	310,294	212,256	197,377	1,086	2	189,623	3,021	3	0.342	0.611	0.893	0.961	0.088
	20/80	584,481	364,630	249,528	239,229	1,078	3	209,659	24,633	5	0.359	0.575	0.840	0.876	0.098
10	80/20	567,681	374,407	309,032	305,332	620	2	294,092	1,704	4	0.518	0.785	0.952	0.963	0.118
	50/50	599,810	312,670	197,448	192,731	2,685	2	180,394	28,936	5	0.301	0.577	0.914	0.936	0.073
	20/80	631,939	371,614	227,774	222,674	1,059	2	197,740	42,652	5	0.313	0.532	0.868	0.888	0.079

traffic is concentrated between the hours of 8:00 A.M. and 5:00 P.M., whereas residential traffic has some activity throughout the day but peaks during the evening hours. In the base problems, the traffic mix is 50% residential and 50% business (denoted 50/50). We also test a mix of 80% residential and 20% business (denoted 80/20) and a mix of 20% residential and 80% business (denoted 20/80) (see Table 5 for the results).

As expected, a 50/50 traffic mix results in the lowest cost per minute, because it has the least variability throughout the day, followed by the 20/80 mix. The cost of the 80/20 mix, dominated by residential traffic, is the most expensive because of the significant variability. The residential busy hour accounts for almost 23% of the traffic, whereas the business busy hour accounts for only 13%. The variability also impacts the savings achieved by a hubbing strategy. For each of the 10 base problems, the cost of H2 relative to NH (H2/NH) increases as the fraction of residential traffic increases. The savings associated with implementing a hubbing strategy are dampened by the variability. Relative to Iter 1, no clear pattern emerges for the smaller problems, but for the larger problems, the H2/I1 ratio also tends to increase as the fraction of residential traffic increases.

The impact of the traffic mix on CPU time is unpredictable, but for most problems, the 20/80 mix is the most difficult. Overall, the 50/50 mix tends to require the fewest iterations. The number of hubs in the H2 solution varies with the traffic mix, although the differences are most pronounced for the larger problems (see Table 7). For these, the fewest number of hubs is used in the 80/20 traffic mix; the increased variability of this mix lowers the number of DS1s that are economical, which, in turn, reduces the opportunity for hubbing savings and results in a higher cost per minute.

Costs. Within H2, key decisions are driven by two cost relationships. First, the number of DS1s leased for DTT depends on the relative cost of DTT versus TST. Second, the optimal hubbing strategy depends, in part, on the relative cost of a DS3 versus a DS1. In our base test problems, we use the averaged costs shown in Table 1. We also consider the impact of

Table 6 Summary of Computational Results: Costs

					H1			H2							
ID	Cost	TST ($)	NH ($)	Iter 1 ($)	Cost ($)	Time(s)	Iter	Cost ($)	Time(s)	Iter	H2/TST	H2/NH	H2/I1	H2/H1	¢/min
1	Base	178,506	115,789	106,594	88,948	1	2	88,634	2	3	0.497	0.765	0.832	0.996	0.149
	7*DS1	178,506	115,789	114,277	113,880	2	2	113,763	3	2	0.637	0.982	0.995	0.999	0.191
	1.2*TST	213,248	124,932	103,399	93,377	16	2	92,488	14	2	0.434	0.740	0.894	0.990	0.156
2	Base	151,527	65,377	55,831	54,295	61	2	52,969	65	3	0.350	0.810	0.949	0.976	0.075
	7*DS1	151,527	65,377	59,766	58,587	155	2	57,215	140	4	0.378	0.875	0.957	0.977	0.081
	1.2*TST	181,458	68,826	57,324	55,087	90	2	52,552	449	4	0.290	0.764	0.917	0.954	0.074
3	Base	190,353	93,068	77,731	76,625	177	2	75,736	233	2	0.398	0.814	0.974	0.988	0.089
	7*DS1	190,353	93,068	82,917	81,808	380	2	81,052	445	3	0.426	0.871	0.978	0.991	0.095
	1.2*TST	227,863	98,410	80,292	79,454	32	2	77,745	110	2	0.341	0.790	0.968	0.978	0.091
4	Base	244,513	149,074	123,502	118,516	84	3	115,879	49	4	0.474	0.777	0.938	0.978	0.131
	7*DS1	244,513	149,074	130,392	128,976	257	3	125,729	426	3	0.514	0.843	0.964	0.975	0.142
	1.2*TST	292,258	158,908	126,659	117,187	86	4	111,722	72	4	0.382	0.703	0.882	0.953	0.126
5	Base	235,383	136,126	93,280	89,330	327	2	85,467	870	3	0.363	0.628	0.916	0.957	0.092
	7*DS1	235,383	136,126	103,226	102,311	1,611	2	99,773	1,601	4	0.424	0.733	0.967	0.975	0.108
	1.2*TST	281,495	142,088	95,618	92,708	260	2	84,468	554	4	0.300	0.594	0.883	0.911	0.091
6	Base	317,376	153,590	125,934	121,193	29	3	118,397	147	3	0.373	0.771	0.940	0.977	0.086
	7*DS1	317,376	153,590	135,040	134,018	170	2	131,299	226	3	0.414	0.855	0.972	0.980	0.095
	1.2*TST	379,827	160,888	126,944	123,244	119	2	122,786	110	2	0.323	0.763	0.967	0.996	0.089
7	Base	350,638	173,805	134,120	131,458	291	2	127,249	2,585	3	0.363	0.732	0.949	0.968	0.084
	7*DS1	350,638	173,805	144,441	142,837	682	2	140,966	678	3	0.402	0.811	0.976	0.987	0.093
	1.2*TST	419,627	183,556	140,296	138,213	302	3	131,386	2,715	4	0.313	0.716	0.936	0.951	0.087
8	Base	401,508	197,789	139,715	129,986	3,026	3	123,663	4,894	4	0.308	0.625	0.885	0.951	0.072
	7*DS1	401,508	197,789	154,245	151,844	10,910	2	144,130	23,346	5	0.359	0.729	0.934	0.949	0.084
	1.2*TST	480,465	206,385	139,360	134,166	9,768	2	126,945	15,954	6	0.264	0.615	0.911	0.946	0.074
9	Base	554,765	310,294	212,256	197,377	1,086	2	189,623	3,021	3	0.342	0.611	0.893	0.961	0.088
	7*DS1	554,765	310,294	233,190	226,585	4,807	3	220,335	11,015	4	0.397	0.710	0.945	0.972	0.103
	1.2*TST	663,374	326,508	214,611	203,294	747	2	197,118	2,942	3	0.297	0.604	0.918	0.970	0.092
10	Base	599,810	312,670	197,448	192,731	2,685	2	180,394	28,936	5	0.301	0.577	0.914	0.936	0.073
	7*DS1	599,810	312,670	220,337	218,551	11,856	3	214,804	22,019	4	0.358	0.687	0.975	0.983	0.086
	1.2*TST	717,599	325,600	201,061	197,954	10,327	3	183,436	49,279	8	0.256	0.563	0.912	0.927	0.074

increasing the DS3 costs and the TST costs (see Table 6 for the results).

First, we consider the impact of increasing both components of the DS3 cost by setting them equal to seven times the DS1 cost. In this case, the DS3 exhibits weaker economies of scale relative to the base costs; as expected, the costs per minute are higher. The DS3 cost has a noticeable effect on the savings achieved by a hubbing strategy. Both ratios H2/NH and H2/I1 increase as the DS3 cost increases, suggesting that H2 is better able to exploit the cost structure under strong economies of scale. Except for Problems 7 and 10, the running times increase with the DS3 costs. The number of iterations varies by one (either way) for most problems. For the smaller problems, increasing the DS3 costs results in a similar number of hubs; for the larger problems, the number of hubs decreases as we increase the DS3 cost.

Second, we consider the impact of increasing both components of the TST cost by 20%. The costs per minute are not always higher relative to the base problems. As TST becomes more expensive, DS1s become economical for more EOs. As a result, we expect the benefits of hubbing to increase. Relative to NH, H2 produces lower cost designs; H2/NH decreases for all problems. The results relative to Iter 1 are mixed. There does not seem to be a clear relationship between CPU time and TST cost nor number of iterations. The number of hubs selected is the same or slightly larger relative to the base solutions, which makes sense. As we increase the TST cost, more EOs

Table 7 Number of Hubs vs. Market Share, Traffic Mix, and Costs

	Market share (%)			Traffic mix (R/B)			Costs		
Problem	10	20	40	80/20	50/50	20/80	7*DS1	Base	1.2*TST
1	2	5	9	6	5	6	4	5	6
2	3	5	11	6	5	6	5	5	6
3	3	6	13	6	6	8	7	6	7
4	3	7	16	8	7	9	6	7	9
5	5	7	14	6	7	9	6	7	9
6	6	11	22	10	11	13	9	11	11
7	6	12	24	11	12	15	10	12	13
8	7	14	26	14	14	16	12	14	14
9	10	16	29	14	16	19	14	16	17
10	11	19	33	13	19	24	16	19	21

Table 8 Comparison of H2 and Linearized Formulation Results

				H2		Linearized formulation		
ID	Mkt (%)	Mix	Cost	Cost ($)	Time(s)	Cost ($)	Time(s)	H2/Linear
1	20	50/50	Base	88,634	2	88,430	131	1.002
	20	80/20	Base	101,269	6	99,149	94	1.021
	20	20/80	Base	96,616	5	96,616	124	1.000
	10	50/50	Base	59,432	1	55,177	101	1.077
	40	50/50	Base	149,798	9	132,763	177	1.128
	20	50/50	7*DS1	113,763	3	100,921	284	1.127
	20	50/50	1.2*TST	92,488	14	92,481	146	1.000
2	20	50/50	Base	52,969	65	51,263	15,536	1.033
	20	80/20	Base	71,900	65	69,088	32,161	1.041
	20	20/80	Base	60,135	266	56,353	14,052	1.067
	10	50/50	Base	31,832	47	31,832	7,369	1.000
	40	50/50	Base	84,052	229	83,211	9,109	1.010
	20	50/50	7*DS1	57,215	140	56,875[a]	86,400	1.006
	20	50/50	1.2*TST	52,552	449	52,449	27,619	1.002
4	20	50/50	Base	115,879	49	107,767	870	1.075
	20	80/20	Base	140,828	102	131,235	1,198	1.073
	20	20/80	Base	124,582	153	117,239	1,743	1.063
	10	50/50	Base	67,415	31	66,950	2,926	1.007
	40	50/50	Base	166,714	120	166,662	2,738	1.000
	20	50/50	7*DS1	125,729	426	125,573	6,914	1.001
	20	50/50	1.2*TST	111,722	72	111,722	2,099	1.000

Note. [a]The linearized formulation was not solved to optimality in 24 hours. The value reported is the best lower bound.

will have DS1s, which will increase the opportunities for hubbing.

Comparison to Linearized Formulation. As we noted in §2, solving the linearized formulation is very difficult. In fact, only the smaller instances can be solved. Table 8 compares the solutions of H2 and the linearized formulation for instances of Problems 1, 2, and 4. For instances of the other problems, the memory requirements were too large. The empirical results indicate that H2 provides high-quality solutions relative to the optimal solution of the linearized version of LDAND. Notice that H2 finds the optimal solution for five of the instances. Overall, the average gap is 3.5% with H2 requiring on average less than two minutes of CPU time and the linearized formulation requiring, on average, almost three hours.

5. Summary and Conclusions

Recent changes in the regulatory environment have created opportunities for long-distance companies to better manage their access networks and to significantly reduce their costs. In this paper, we have studied a problem that arises in the design of access networks for long-distance communications in the United States. We have developed a novel three-phase approach that considers the stochastic aspects of the problem. Our computational results indicate a potential cost savings of hundreds of millions of dollars annually to long-distance companies. Although we have described the problem in terms of a managerial problem specific to the U.S. telecommunications industry, we believe that our approach has relevance beyond the United States as deregulation of the telecommunications industry spreads worldwide.

With the continued advances in communications technology, local telephone companies are beginning to offer higher-capacity optical facilities. Our understanding is that, so far, these facilities generally are available and used for data traffic. However, in the future, if a long-distance company decides to lease higher-capacity optical facilities for DTT, the LDAND problem can be modified to handle the additional layer of multiplexing and the additional facility types. Preliminary experiments indicate that our approach is viable with more than two facility types.

In this paper, we have assumed that all traffic is carried. In practice, however, a small fraction of traffic may be blocked. To model the blocking, we would need a characterization of the traffic of the other long-distance companies sharing the trunks connecting the tandem switch to each EO as well as the number of trunks provisioned by the local telephone company. These data elements are not available to a long-distance company as they are considered confidential information. Consequently, we do not model this blocking in the LDAND problem. Further, since the blocking probability is extremely small (generally on the order of 0.05% during the busy period for long-distance traffic), we do not believe that it will materially affect the results.

To conclude, with the (global) deregulation in the telecommunications industry, the opportunities for applying management science techniques to the management of communications networks are increasing. Our research demonstrates a novel method for solving a new and a high-impact network management problem in the telecommunications industry that can produce hundreds of millions of dollars in cost savings annually for long-distance companies.

Appendix

A Bound for s_{MAX_i}. As noted earlier, the number of DTT DS1s that prove in at an EO is a function of the cost of leasing DS1s from the EO to its permissible hubs. A priori, we do not know the best hubbing strategy, so we compute an upper bound on the number of DS1s at EO $i \in N$, i.e., s_{MAX_i}. (We use s_{MAX_i} in the approach outlined in §2.3.) Recall from *Initial Values* that we add DS1s while the decrease in TST costs is larger than the cost of adding a DS1. Therefore, for EO i, an upper bound on the number of DS1s is the smallest j for which k_{ji}, the decrease in TST costs associated with adding a jth DS1, is less than the smallest possible DS1 cost. The smallest possible DTT DS1 cost is the cost of a DS1 when an EO is assigned to its closest permissible hub.

$\mathcal{NP}$-Completeness Proof. It is well known that the standard transportation problem can be transformed to a minimum cost flow problem on a bipartite graph. Thus, to prove $\mathcal{NP}$-completeness, it suffices to consider the transportation problem with the additional constraint that a supply node may supply at most e demand nodes. Note, due to the symmetric nature of the transportation problem (just reverse the role of supply and demand nodes), the degree constraints equivalently could be on the demand nodes (i.e., a demand node may be supplied by no more than e supply nodes). We call this the degree-constrained transportation problem.

The following result, described to us by Zhi-Long Chen (2003) at University of Maryland, shows that even the feasibility version of the degree-constrained transportation problem is $\mathcal{NP}$-complete.

Theorem 2. *The degree-constrained transportation problem, with degree equal to one or three, is $\mathcal{NP}$-complete.*

Proof. Consider the $\mathcal{NP}$-complete 3-partition problem. Given $3m$ items, each with integer size b_i with $(1/4)b < b_i < (1/2)b$ and $mb = \sum b_i$. The decision version asks: Is there a partition of these $3m$ items into m disjoint subsets each with exactly three items and a total size b. We transform the 3-partition problem to an instance of the degree-constrained transportation problem as follows. Use the $3m$ items as $3m$ supply nodes, each with supply amount b_i. They will supply m demand nodes each with a demand b. The transportation costs are zero. This corresponds to an instance of the feasibility version of the degree-constrained transportation problem with the constraint that each supply node supply no more than one demand node. (A feasible solution provides a yes answer and an infeasible solution provides a no answer.) Or, equivalently, it corresponds to an instance of the feasibility version of the degree-constrained transportation problem with the constraint that each demand node be supplied by no more than three supply nodes. Thus, the degree-constrained transportation problem with degree equal to 1 or 3 is $\mathcal{NP}$-complete. □

References

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Balakrishnan, A., T. L. Magnanti, A. Shulman, R. T. Wong. 1991. Models for planning capacity expansion in local access telecommunication networks. *Ann. Oper. Res.* **33** 239–284.

Boorstyn, R. B., H. Frank. 1977. Large-scale network topological optimization. *IEEE Trans. Comm.* **25** 29–47.

Chen, Z.-L. 2003. Personal e-mail communication.

FCC. 1999. Fifth report and order and further notice of proposed rule making. Technical report FCC99-206, Federal Communications Commission, Washington, D.C.

FCC. 2001. Statistics of the long distance telecommunications industry. Technical report, Federal Communications Commission, Industry Analysis Division, Common Carrier Bureau, Washington, D.C.

Frantzeskakis, L. F., H. Luss. 1999. The network redesign problem for access telecommunications networks. *Naval Res. Logist.* **46** 487–506.

Freeman, R. L. 1996. *Telecommunication System Engineering*. John Wiley and Sons, New York.

Gavish, B. 1991. Topological design of telecommunication networks—Local access design methods. *Ann. Oper. Res.* **33** 17–71.

Lande, J., K. Lynch. 2002. Telecommunications industry revenues 2000. Technical report, Federal Communications Commission, Industry Analysis Division, Common Carrier Bureau, Washington, D.C.

McGregor, P., D. Shen. 1977. Network design: An algorithm for the access facility location problem. *IEEE Trans. Comm.* **25** 61–73.

Messerli, E. J. 1972. Proof of a convexity property of the Erlang B formula. *Bell System Tech. J.* **51** 951–953.

Nemhauser, G. L., L. A. Wolsey. 1999. *Integer and Combinatorial Optimization*. John Wiley and Sons, New York.

Salman, F. S., R. Ravi, J. Hooker. 2001. Solving the local access network design problem. Technical report, Krannert School of Management, Purdue University, West Lafayette, IN.

Sherali, H. D., Y. Lee, T. Park. 2000. New modeling approaches for the design of local access transport area networks. *Eur. J. Oper. Res.* **127** 94–108.