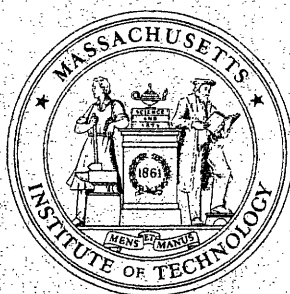


# OPERATIONS RESEARCH CENTER

working paper



**MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY**



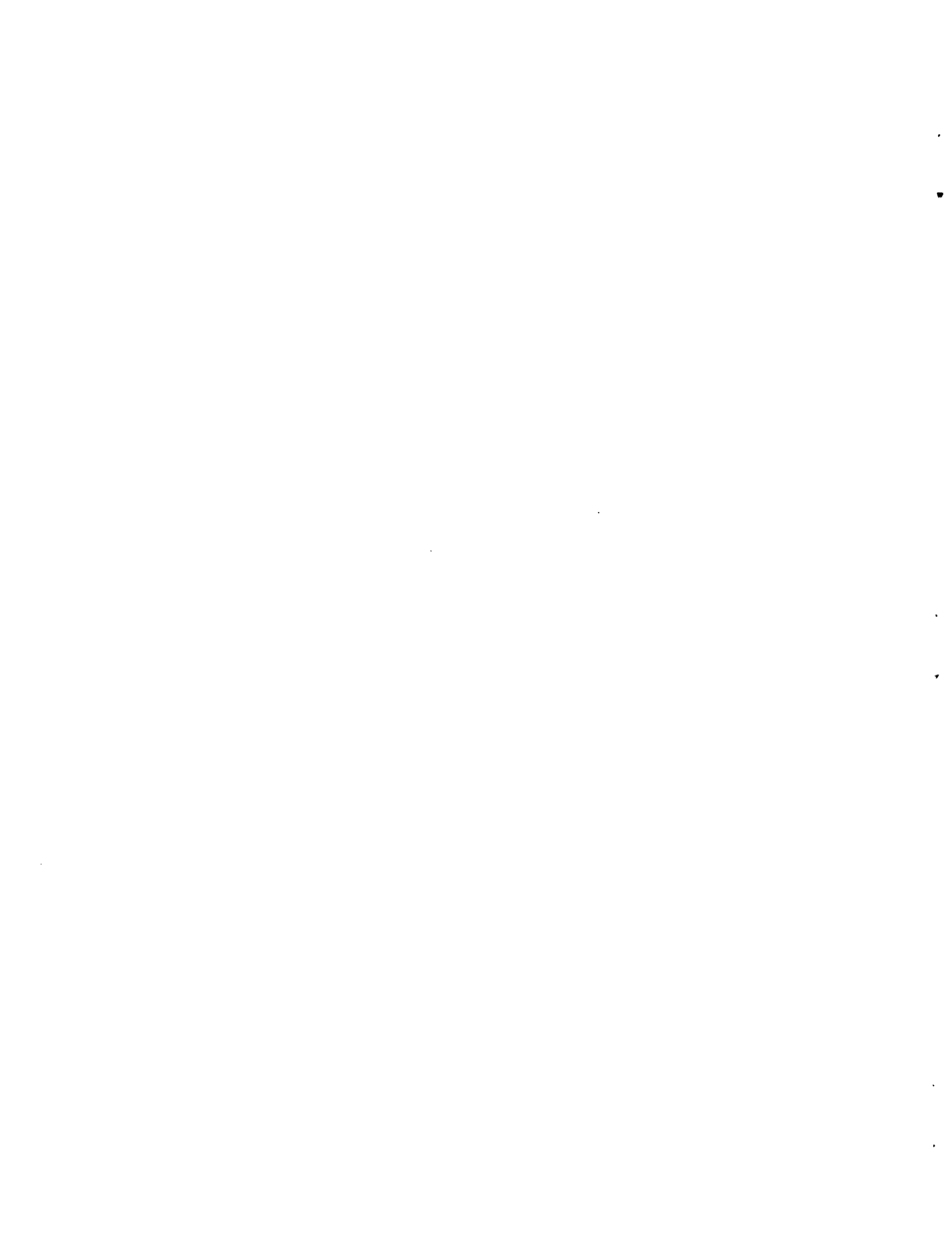
Designing Satellite Communication  
Networks by Zero-One Quadratic  
Programming

by

Marcia P. Helme and Thomas L. Magnanti

OR 159-87

January 1987



**Designing Satellite Communication  
Networks by Zero-One Quadratic  
Programming**

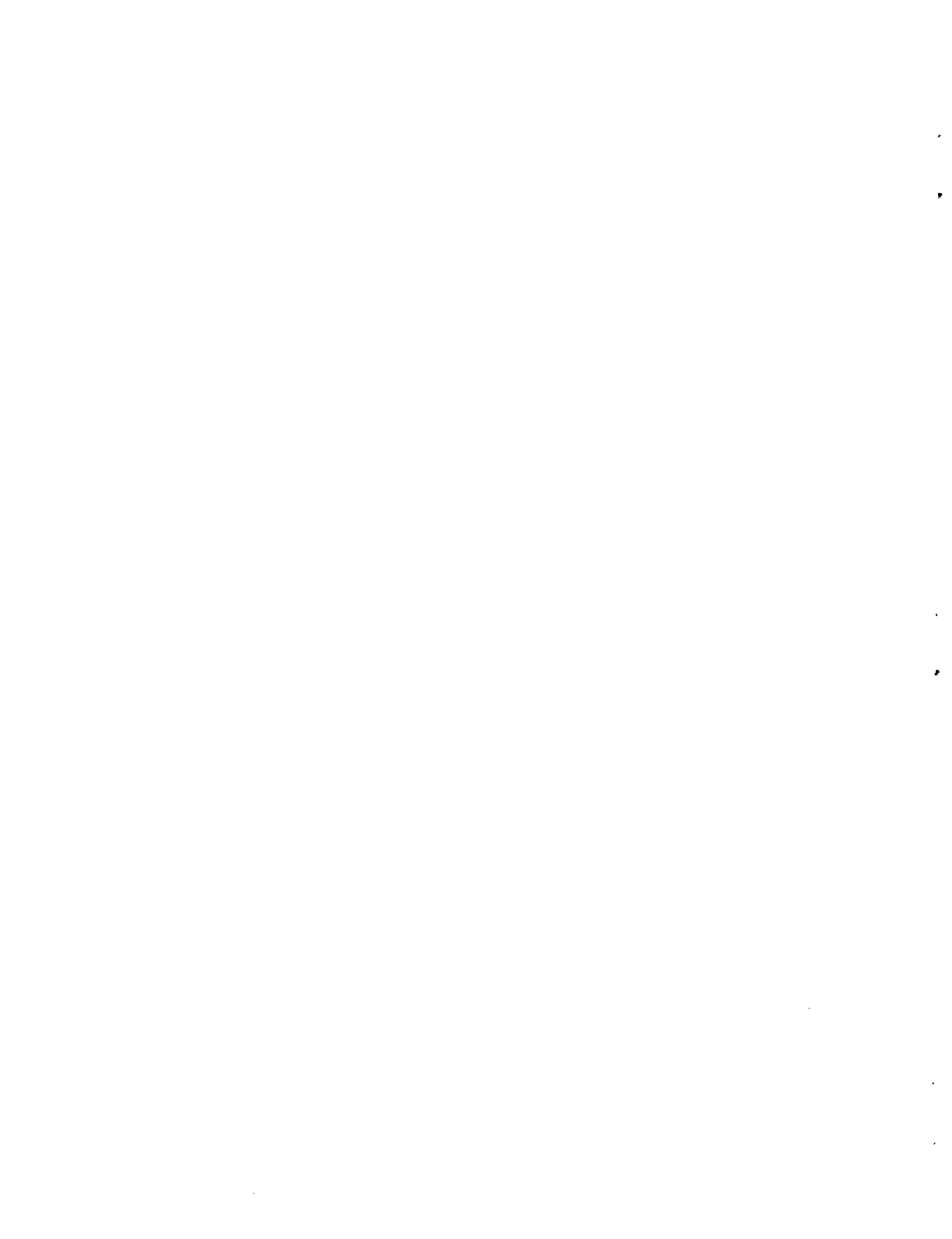
**by**

**Marcia P. Helme  
GTE Laboratories, Inc.**

**Thomas L. Magnanti<sup>\*</sup>  
Sloan School of Management  
M.I.T.**

**January, 1987**

\* Supported in part by Grant #ECS-83-16224 from The Systems Theory and Operation Research Program of the National Science Foundation.



## **ABSTRACT**

In satellite communications networks, distinctive facilities called homing stations perform special transmission functions. Local demand nodes clustered around each homing station communicate with each other via a local switch at the homing station; demand nodes in different clusters communicate with each other via satellite earth stations at the homing stations. Designing such a communication network requires choices on the locations of the earth stations and on the assignments of demand nodes to the local clusters at the earth stations. We formulate this problem as a zero-one quadratic facility location problem and transform it into an equivalent zero-one integer linear program. Computational experience on real data shows that a branch and bound procedure is effective in solving problems with up to forty demand nodes (major cities) and that the solutions that this algorithm finds improve considerably upon management generated solutions. We also show that a greedy add heuristic, as implemented on an IBM PC, consistently generates optimal or near-optimal solutions.





## 1. INTRODUCTION.

The topological design of a voice carrying satellite communication network is a special type of facility location problem. The problem embodies decisions on how to connect pairs of nodes in a communication network by a combination of terrestrial and satellite links and special communication facilities, called homing stations, at the nodes. Each homing station consists of an earth station and a local switch. The earth stations communicate with each other through the satellite, and the terrestrial links connect each non-homing station node to a homing station (see Figure 1). Nodes connected to the same homing station communicate through the local switch, while nodes connected to different homing stations use their assigned earth stations and the satellite. The following decisions must be made:

1. At which nodes should homing stations be placed?
2. To which homing station should a non-homing station node be assigned?
3. What should be the capacity of the earth stations, local switches, and terrestrial links?

We assume the following data are known: (1) the location of each node, (2) the demands for service for each pair of nodes, and (3) the cost parameters. We address the problem of finding the minimum cost configuration that meets all demands for service.

This problem is becoming increasingly important in practice. The demand for long haul communication services is growing in response to the recently deregulated competitive communications environment. Satellites have several well known advantages as a telecommunications medium. First, the capital costs are distance insensitive. Second, satellites can carry all types of telecommunication traffic. Third, new long haul links are easy to add to an existing satellite network: installation of the  $(n + 1)$ st earth station immediately provides  $n$  new long haul links, a distinct advantage in crowded urban areas. Furthermore,

plenty of satellite transponder capacity is currently available: an estimated forty percent of the available transponder capacity is not in use (*Telecommunications*, June, 1985).

The analysis in this paper was motivated by a real world problem of providing long haul voice communication service. As described in Section V, the optimization model has been implemented with real data on networks of up to forty nodes.

A satellite communication network to serve voice traffic has three important characteristics that impact its topological design. First, the cost of long haul traffic traveling between two earth stations by means of a satellite is independent of the distance between those earth stations. Second, the demands for voice communication service occur between pairs of nodes. Third, the demands for voice service are not uniform. For example, the projected traffic between New York and Los Angeles is much heavier than the projected traffic between Little Rock and Schenectady. By contrast, in a data communication network, the demands for service frequently occur at each single node, which is to communicate with a central facility; moreover, models of data communication networks often assume the demands are uniform.

In this paper we formulate the satellite network problem as a zero-one integer quadratic programming model (IQP) with an objective function that is neither convex nor concave. We also present a zero-one linear programming (ILP) model that is equivalent to the IQP, but has an enlarged set of variables and constraints. We then describe a branch and bound solution method and, using real data, compare its solutions with those found by a greedy heuristic and with those generated by management. Our experience indicates that the branch and bound procedure is very effective in solving problems with up to forty demand nodes and that one version of a greedy heuristic, as implemented in BASIC on an IBM PC, consistently finds optimal or near-optimal solutions.

The integer programming literature on plant location problems is voluminous and has a long history. The first formulations originated with Balinski and Wolfe [4], Kuehn and Hamburger [23], Manne [29] and Stollsteimier [40]. Books by Francis and White [16] and

Handler and Mirchandani [19] contain a wealth of citations to the literature. More recently Tanenbaum [41] used a  $p$ -median problem formulation to choose the location of concentrators on a telecommunication network. This formulation seeks  $p$  facility locations that minimize the weighted sum of the distances between each node and its closest facility. Tansel, Francis and Lowe [42] have summarized results on facility location on a network, and a forthcoming book edited by Francis and Mirchandani [15] will give a comprehensive status report of the field. Magnanti and Wong [28] describe work on the closely related network design problem.

Effroymsen and Ray [12] proposed a branch and bound procedure for the plant location problem; this approach has subsequently been improved upon by many authors including Sa [36], Davis and Ray [11], Khumawala [22], Akinc and Khumawala [1], and Nauss [33]. Spielberg [39] proposed a direct search scheme. Many other authors have proposed and studied dual-based procedures: Bilde and Krarup [5], Curnuejols, Fisher and Nemhauser [9], Erlenkotter [13], Nauss [33], Guignard and Spielberg [18], and Christofides and Beasley [8]. In addition, several other authors have considered capacitated versions of this problem. See Wong's [43] annotated bibliography for citations to this literature.

Several researchers have studied the multi-level hierarchy homing station location problem. For example, see Boorstyn and Frank [7], Mendelsohn, Boorstyn and Kershenbaum [31], Gavish [17], Kershenbaum and Boorstyn [21], and Mirzaian [32]. Boorstyn and Frank [7] present a heuristic approach for deciding which nodes should be homed into which homing stations. Mirzaian [32] uses both Lagrangian relaxation and an approximation algorithm on an integer linear programming formulation of the problem. These authors focus on data communication networks. They assume uniform demands between nodal pairs, and their cost function does not address the sizes of the links and homing stations. The Kershenbaum and Boorstyn [21] minimal spanning tree approach allows nonuniform demands, but assumes that demands are created by nodes rather than by nodal pairs.

There is a tendency to formulate the satellite network problem as a traditional transportation network facility location problem with known demands at the nodes and with lin-

ear costs. In principle, this perspective would permit us to build upon the linear cost facility location literature cited in this section. In the traditional transshipment formulation, the nodal demands  $v(i)$  are found from the pairwise nodal demands  $v(i,j)$  by defining  $v(i)$  as

$$v(i) = \sum_{j=1}^N v(i,j).$$

(We assume  $v(i,i) = 0$ .) Figure 2 depicts such a formulation with four candidate locations  $\{1,2,3,4\}$  for homing stations. The heavy lines show a feasible solution.

Figure 2, however, does not accurately depict the problem. It incorrectly assumes that each earth station supplies all the nodes homing to it, and it does not recognize that the within cluster traffic bypasses the satellite. If nodes 7,8, and 9 home to node 3 in the optimal solution, as shown in Figure 2, then they will use the satellite to communicate with nodes 1,2,4,5,6,10, and 11; but they will communicate with each other by a local switch at node 3, not by the satellite. Therefore, the earth station at node 3 does not fully supply all the nodes attached to it.

The transshipment formulation will be deficient whenever the long haul transmission cost and the local switch cost differ substantially. If the per circuit long haul cost significantly exceeds the per circuit cost of a local switch, then homing neighboring high volume nodal pairs into the same homing station permits traffic to bypass the expensive long haul medium. We will show that this potential for synergy among neighboring nodes introduces quadratic terms into the objective function. Consequently, the satellite network application requires us to depart from the traditional linear cost facility location literature.

To close this section, we note that the satellite communication network problem is closely related to other problems involving point-to-point delivery systems: e.g., less than truckload freight distribution (Powell and Sheffi [34], Lamar, Sheffi and Powell [24], and Balakrishnan and Graves [3]), rail freight planning (Bodin, Golden, Schuster and Romig [6], and Assad [2]), and package delivery (Leung, Magnanti and Singhal [26], Singhal [38], and Leung [25]).

## 2. FORMULATION.

### 2.1 PROBLEM INGREDIENTS.

We assume that the communication system has  $N$  nodes, each with a known location. A node might represent a single city or an aggregate of several cities with a selected geographic center. We also assume that each pair of nodes has a known demand for two-way communication, and that all the necessary cost parameters are known. We seek the least cost network configuration that fully connects all nodes and meets all demands for service.

### 2.2 TERMINOLOGY.

A feasible solution to the model is a partitioning of the nodes into disjoint sets, called clusters. Each cluster contains one node called a *homing station*, or *major node*, that consists of a local switch and an earth station. The remaining nodes in the cluster, called *minor nodes*, attach to the homing station by terrestrial links.

Figure 1 depicts a feasible solution with two clusters.

Any solution divides the traffic into two distinct types: within cluster traffic and between cluster traffic. Nodal pairs within the same cluster, such as nodes 1 and 2, and nodes 3 and 7, in Figure 1 generate within cluster traffic that requires the use of only terrestrial links and the local switch for that cluster; this traffic uses neither the satellite nor the earth stations. Nodal pairs not in the same cluster, such as nodes 1 and 3, and nodes 5 and 7, generate between cluster traffic that requires the use of the satellite. Traffic originating at node 1 and destined for node 3 travels from node 1 to node 5 along a terrestrial link, then from node 5 to node 6 by use of the satellite, and finally from node 6 to node 3 by a terrestrial link.

In telecommunication cost modeling, the distinction between these types of traffic is desirable whenever the cost structures differ. The cost of satellite communication between two earth stations does not depend upon their distance from each other, while the cost of terrestrial communication does.

### **2.3 ASSUMPTIONS.**

We make the following simplifying assumptions.

1. In every feasible solution, every node attaches to exactly one major node. A node that becomes a major node is considered to attach to itself.
2. If node  $i$  is attached to major node  $k$ , then a terrestrial link between  $i$  and  $k$  carries node  $i$ 's traffic to the rest of the network. If  $i=k$ , then the link has length zero.
3. One satellite, with infinite capacity, is accessible to any potential earth station in the network.

These assumptions imply that every feasible solution has the following characteristics.

- i) Every cluster has a star configuration.
- ii) Every nodal pair has a single communication path; there are no alternate routes.

## 2.4 THE INTEGER QUADRATIC PROGRAM (IQP).

We define the following notation for the known quantities.

$N$  = number of nodes.

$v(i,j)$  = volume of demand, in circuit requirements, for traffic between nodes  $i$  and  $j$ . Since  $i$  and  $j$  will communicate with each other, in both directions, along the same communication path, we may assume that, for  $j > i$ ,  $v(i,j)$  denotes the total bidirectional traffic. Therefore, we set  $v(i,j) = 0$  whenever  $i \geq j$ .

$v(i)$  = demand for traffic originating or terminating at node  $i$ ;

$$\text{thus, } v(i) = \sum_{j=1}^N v(i,j) + \sum_{j=1}^N v(j,i).$$

$a(k)$  = fixed cost for establishing a homing station at node  $k$ .

$b$  = per circuit cost of capacity of an earth station.

$c(i,k)$  = per circuit mile cost of a terrestrial link homing minor node  $i$  to major node  $k$ ;  $c(i,i) = 0$ .

$d$  = per circuit cost of capacity of a local switch. We assume  $b > d$ .

To reduce the number of constraints and variables, we establish a subset of the nodes as candidates for homing station locations. This candidate set can reflect certain practical considerations. For example, if a terrestrial network is currently in place, with central offices acting as homing stations to which local links are already attached, then the

candidate set might include nodes representing these central offices. We also wish to eliminate from consideration any obviously unattractive solutions, such as attaching the Seattle node to a homing station at Miami. Thus, for each node, we define a set of candidate homing stations to which it can attach. By proper indexing, we can thus define the following sets.

$S = \{1,2,\dots,N\}$  = set of all nodes;

$S' = \{1,2,\dots,m\}$  = set of nodes that are candidates to become homing stations;  $m \leq N$ ; and

$S(i)$  = set of candidate homing stations to which node  $i$  can attach;

We define the decision variables  $x(i,k)$  as follows.

For  $k = 1,2,\dots,m$ ,

$x(k,k) = 1$  if a homing station is established at node  $k$ ,  
 $= 0$  otherwise.

For  $i = 1,\dots, m$  and  $k \in S(i)$ ,

$x(i,k) = 1$  if node  $i$  attaches to homing station  $k$ ,  
 $= 0$  otherwise.

The variable definitions and our previous assumptions produce the following constraints.

$$\sum_{k \in S(i)} x(i,k) = 1 \text{ for } i = 1,2,\dots,N.$$

$$x(i,k) \leq x(k,k) \text{ for } i = 1,2,\dots,N; k \in S(i); k \neq i.$$

$$x(i,k) = 0,1 \text{ for } i = 1,2,\dots,N; k \in S(i).$$



The first set of constraints ensures that every node either becomes a homing station or attaches to one. The second set of constraints prohibits a node from homing into any node that does not contain a homing station. The final, binary restrictions force each node to attach to exactly one major node.

We now express the two types of traffic in any feasible solution in terms of the decision variables  $x(i,k)$ . First, we introduce the following set notation.

$R$  = set of major nodes in a given solution.

$R(k)$  = set of nodes homing to major node  $k$ , including  $k$  itself.

$S(i,j) = S(i) \cap S(j)$ .

The total traffic is given by

$$T_{TOT} = \sum_{i=1}^N \sum_{j=1}^N v(i,j).$$

The within cluster traffic is

$$\begin{aligned}
 T_W &= \sum_{k \in R} \sum_{i \in R(k)} \sum_{j \in R(k)} v(i,j) = \sum_{k=1}^m \sum_{i=1}^N \sum_{j=1}^N v(i,j) x(i,k) x(j,k) \\
 &= \sum_{i=1}^N \sum_{j=1}^N v(i,j) \sum_{k \in S(i,j)} x(i,k) x(j,k).
 \end{aligned}$$

Therefore, the between cluster traffic is

$$T_B = T_{TOT} - T_W = \sum_{i=1}^N \sum_{j=1}^N v(i,j) \left[ 1 - \sum_{k \in S(i,j)} x(i,k) x(j,k) \right].$$

We are now ready to express the cost of any feasible solution. It consists of four parts.

- i) fixed homing station cost:  $\sum_{k=i}^m a(k) x(k,k)$ .
- ii) capacity dependent earth station cost:  $2 * b * T_B$ .

The factor of 2 in this expression conforms with our earlier observation that two earth stations are used for the satellite medium portion of between cluster communication; in contrast, local traffic, considered in item (iv) to follow, uses only one local switch.

iii) terrestrial link cost:

$$\begin{aligned} \sum_{k \in R} \sum_{i \in R(k)} v(i) * c(i,k) &= \sum_{k=1}^m \sum_{i=1}^N v(i) * c(i,k) * x(i,k) \\ &= \sum_{i=1}^N v(i) \sum_{k \in S(i)} c(i,k) * x(i,k). \end{aligned}$$

We note that whenever  $i \in R(k)$ , link  $(i,k)$  carries all of node  $i$ 's traffic.

iv) capacity dependent local switch cost:

$$d * T_W.$$

Recall that, as contrasted with (ii), only one local switch is used for within cluster communication.

Thus, the total cost is

$$\begin{aligned} &\sum_{k=1}^m a(k) * x(k,k) + 2b \sum_{i=1}^N \sum_{j=1}^N v(i,j) \\ &- 2b \sum_{i=1}^N \sum_{j=1}^N v(i,j) \sum_{k \in S(i,j)} x(i,k) * x(j,k) + \sum_{i=1}^N v(i) \sum_{k \in S(i)} c(i,k) * x(i,k) \\ &+ d \sum_{i=1}^N \sum_{j=1}^N v(i,j) \sum_{k \in S(i,j)} x(i,k) * x(j,k). \end{aligned}$$

Collecting together the problem ingredients that we have discussed gives the following zero-one quadratic programming model.

Minimize

$$\begin{aligned} & \sum_{k=1}^m a(k) \cdot x(k,k) + \sum_{i=1}^N v(i) \sum_{k \in S(i)} c(i,k) \cdot x(i,k) \\ & - (2b-d) \sum_{i=1}^N \sum_{j=1}^N v(i,j) \sum_{k \in S(i,j)} x(i,k) \cdot x(j,k) \end{aligned} \quad (1)$$

subject to

$$\sum_{k \in S(i)} x(i,k) = 1 \quad \text{for } i = 1, 2, \dots, N \quad (2)$$

$$x(i,k) \leq x(k,k) \quad \text{for } i = 1, 2, \dots, N; k \in S(i); k \neq i \quad (3)$$

$$x(i,k) = 0, 1 \quad \text{for } i = 1, 2, \dots, N; k \in S(i) \quad (4)$$

The objective function for this quadratic integer program is neither convex nor concave; the matrix associated with the quadratic terms has a special structure: it is block diagonal with zero main diagonal. Figure 3 provides an example of this matrix for which  $S(3) = S(4) = S(5) = \{1, 2\}$ . In the next subsection we develop a zero-one integer linear program as a solvable surrogate for the quadratic integer program.

## 2.5 THE INTEGER LINEAR PROGRAM (ILP).

Possibly the most natural approach to formulating the quadratic integer program as an integer linear program would be to replace the quadratic terms  $x(i,k)*x(j,k)$  in the objective function (1) by new zero-one variables  $y(i,j,k)$ , and to define linear constraints of the form

$$\begin{aligned} y(i,j,k) &\leq x(i,k) \\ y(i,j,k) &\leq x(j,k). \end{aligned} \tag{5}$$

Since  $b > d$ , it is not necessary to require the  $y$  variables to be zero-one. If all the  $x$  variables are zero-one, then all the  $y$  variables become zero-one because of their negative objective function coefficient which will cause each  $y$  to take on its maximum value at optimality. The structure of the problem will, however, permit us to use fewer new variables, all of which are continuous.

We define the zero-one variables  $w(i,j)$  for all  $(i,j)$  with  $j > i$  and  $S(i,j) \neq \emptyset$  as follows:

$$w(i,j) = \sum_{k \in S(i,j)} x(i,k)*x(j,k) \tag{6}$$

and we replace the quadratic terms in the objective function with the variables  $w(i,j)$ . For  $i = 1,2,\dots,N$ ,  $j = 1,2,\dots,N$ , and  $j > i$ ,  $w(i,j)$  can be interpreted as

$$\begin{aligned} w(i,j) &= 1 \text{ if nodes } i \text{ and } j \text{ are in the same cluster.} \\ &= 0 \text{ otherwise.} \end{aligned}$$

We note that (6) can be written in the following form if (2) is satisfied and all the variables are zero-one.

$$w(i,j) = 1 - \max_k \{ |x(i,k) - x(j,k)| \}. \tag{7}$$

In this expression,  $x(i,k) \equiv 0$  if  $k \notin S(i)$  and  $x(j,k) \equiv 0$  if  $k \notin S(j)$ . To obtain a linear integer programming formulation, we replace each equation of type (7) with the following constraints:

$$\left. \begin{aligned} w(i,j) &\leq 1 + x(i,k) - x(j,k) \\ w(i,j) &\leq 1 - x(i,k) + x(j,k) \end{aligned} \right\} \begin{array}{l} \text{for } i = 1, 2, \dots, N \\ j = 1, 2, \dots, N \\ j > i; k \in S(i,j). \end{array} \quad (8)$$

Note that (8) is not equivalent to (7), since the righthand side of (7) might equal one while the lefthand sides of (8) are zero for all  $k$ . However, for  $b > d$  or even  $b > d/2$ , the objective function's coefficients of all  $w(i,j)$  are negative, so that each  $w(i,j)$  will be at its maximum possible value at optimality.

Notice that to limit the number of constraints in the problem formulation, we have not written the set of inequalities (8) for all  $k = 1, 2, \dots, m$ . Therefore, one additional set of constraints must be added to the formulation, namely

$$w(i,j) \leq \sum_{k \in S(i,j)} [x(i,k) + x(j,k)] \quad \text{for } i = 1, 2, \dots, N; j = 1, 2, \dots, N; j > i. \quad (9)$$

Without (9), the following choice would be possible for a given  $i$  and  $j$ ,

$$\begin{aligned} x(i,k) &= x(j,k) = 0 \text{ for all } k \in S(i,j) \\ x(i,p) &= 1 \text{ for a given } p \in S(i), p \notin S(j), \\ x(j,q) &= 1 \text{ for a given } q \in S(i), q \in S(j), \text{ and} \\ w(i,j) &= 1. \end{aligned}$$

Three issues must be considered regarding the clustering variables  $w(i,j)$ . First, the constraints (8) and (9) are *sufficient* to meet their intended objective, i.e., they force  $w(i,j) = 1$  if  $i$  and  $j$  are in the same cluster. Second, both (8) and (9) are *necessary* i.e., without both

of them, the  $w(i,j)$  might not attain the necessary values. Third, (8) and (9) are *consistent* i.e., when  $w(i,j)$  is forced to be a certain value by one set of constraints, it is not simultaneously forced to be a different value by the other set of constraints.

Note that if we establish all of these requirements, then we will have demonstrated that the integer requirements on the  $w(i,j)$  variables are unnecessary. That is, suppose that we merely require

$$0 \leq w(i,j) \leq 1,$$

and impose integrality only on the variables  $x(i,k)$ . If nodes  $i$  and  $j$  belong to different clusters, the constraints (8) and (9) ensure that

$$w(i,j) = 0.$$

If  $i$  and  $j$  belong to the same cluster, the constraints (8) and (9) impose the restriction

$$w(i,j) \leq 1,$$

and the objective function ((10) to follow) forces the  $w(i,j)$  to its maximum value, i.e.,

$$w(i,j) = 1.$$

Table I demonstrates all three requirements. There are two cases: in case 1 nodes  $i$  and  $j$  are in the same cluster, and in case 2 they are not. Case 2 has three subcases: in subcase 2a,  $i$  and  $j$  each home to a station in the other's candidate set; in subcase 2b only one of them homes to a station in the other's candidate set; and in subcase 2c neither of them do. More specifically, we write

case 1:  $k \in S(i), k \in S(j), k \in S(i,j), l \in S(i,j), p \in S(i,j)$   
 case 2a:  $k \in S(i), l \in S(j), k \in S(i,j), l \in S(i,j), p \in S(i,j)$   
 case 2b:  $k \in S(i), l \in S(j), k \in S(i,j), \cancel{l \in S(i,j)}, p \in S(i,j)$   
 case 2c:  $k \in S(i), l \in S(j), \cancel{k \in S(i,j)}, \cancel{l \in S(i,j)}, p \in S(i,j)$ .

Note that constraint (8-k), i.e., (8) written for major node k, is not part of the formulation unless  $k \in S(i,j)$ .

Assuming that (2) holds, we see that the entries in Table I are sufficient to represent these two cases. In the table, the variables  $x(i,p)$  and  $x(j,p)$  represent assignments to a cluster p that contains neither i nor j.

The table demonstrates *sufficiency* because the  $w(i,j)$  assume the desired values in each case, i.e., the value of  $w(i,j)$  obtained from the model is the same as the desired value. The table also shows the *necessity* of both conditions (8) and (9): (8) is needed in case 1 to force

$$w(i,j) \leq 1;$$

(8) is needed in cases 2a and 2b to force

$$w(i,j) = 0;$$

and (9) is needed in case 2c to force

$$w(i,j) = 0.$$

We do not know in advance which of these cases will apply. Thus both (8) and (9) are needed. Finally, the table demonstrates *consistency* of the constraints by showing that the resulting righthand side values are consistent within each of the four cases.



The resulting zero-one mixed integer linear program is:

Minimize

$$\begin{aligned} & \sum_{k=1}^m a(k) \cdot x(k,k) + \sum_{i=1}^N v(i) \sum_{k \in S(i)} c(i,k) \cdot x(i,k) \\ & - (2b-d) \sum_{i=1}^N \sum_{j=1}^N v(i,j) \cdot w(i,j) \end{aligned} \tag{10}$$

subject to

$$(2), (3) (4), (8), (9) \text{ and } (11)$$

where (11) is given by

$$w(i,j) \geq 0. \tag{11}$$

Note that linearizing the quadratic problem with the  $w(i,j)$  variables, as opposed to the more straightforward  $y(i,j,k)$  variables, decreases the number of variables and increases the number of constraints. The question arises as to whether this choice is beneficial, keeping in mind that a linear program's solveability is typically dominated more by the number of constraints than by the number of variables. Thus short wide coefficient matrices are usually preferred to square matrices. We first define the following size related variables:

$$M = \sum_{i=1}^N |S(i)|;$$

$Q$  = number of nonempty  $S(i,j)$ ,

$L$  = number of  $(i,j,k)$  triplets with  $k \in S(i,j)$ .

Then

$M$  = number of  $X$  variables,

$Q$  = number of  $W$  variables,

$$L = \sum_{(i,j)} |S(i,j)| = \text{number of } Y \text{ variables.}$$

Note that  $Q < L$ , and that the denser the network, as measured by the size  $M/N$  of an average  $S(i)$ , the greater the disparity between  $Q$  and  $L$ . The zero-one mixed integer linear program has the following dimensions:

$M$  zero-one variables,

$Q$  continuous variables,

$N + (M-m) + 2L + Q$  constraints.

Using the  $y(i,j,k)$  variables would produce a mixed integer linear program with the following dimensions:

$M$  zero-one variables,

$L$  continuous variables,

$N + (M-m) + 2L$  constraints.

The  $w(i,j)$  variables therefore produce a tall, thin primal linear program matrix and thus a short, wide dual matrix. The  $y(i,j,k)$  variables produce a primal matrix that is squarer; moreover, both its primal and its dual contain more rows than the dual of the  $w(i,j)$  formulations. The denser the network, the greater this disparity and therefore the greater the apparent advantage of the  $w(i,j)$  formulation to obtain quicker solutions to linear programming relaxations of the problem.

### 3. RELAXED FORMULATION AND AN UPPER BOUND.

Although the integer linear programming formulation of the quadratic satellite communication network problem permits us to use linear programming based branch and bound as a solution procedure, this formulation greatly increases the number of variables and constraints. As a further complication, the ILP formulation has a fairly large number of integer variables,  $x(i,k)$  and  $x(k,k)$ .

In this section, we consider a relaxed version (RILP) of the integer linear program, obtained from the ILP by relaxing the integer constraints of the variables  $x(i,k)$  for  $i \neq k$ . Therefore, in the RILP, only the  $m$  variables  $x(k,k)$  must be integer. This relaxation reduces the number of integer variables from  $M$  to  $m$ . The applications considered in the paper produce a reduction factor ( $M/m$ ) of from 6 to 9 as shown in Table II. We will show that for any feasible solution  $(X^0, W^0)$  to the relaxed problem, there is an easily computed feasible solution  $(X^1)$  to the integer quadratic problem (IQP) formulation (1) - (4), with the property that

$$f(X^1) \leq f(X^0),$$

where  $f$  is the objective function of the IQP. Consequently, by requiring only a small subset of the variables to be integer, we can obtain an all integer feasible solution. This fact permits us, in principle, to generate a potentially cost effective feasible solution by solving a much smaller integer program. (We did not, however, use this approach in our computational studies.)

Before establishing this result, let us summarize several facts that are easy consequences of our previous development. In this discussion, let  $g$  denote the objective function of the ILP as a function of its decision variables  $(X^0, W^0)$ .

- F1: If  $(X^0, W^0)$  is feasible for the ILP, then  $X^0$  is feasible for the IQP, and  $g(X^0, W^0) = f(X^0)$ .

F2: If  $X^0$  is feasible for the IQP, then there is a unique  $W^0$  with the property that  $(X^0, W^0)$  is feasible for the ILP, and  $g(X^0, W^0) = f(X^0)$ .

F3: If  $(X^0, W^0)$  is optimal for the RILP and is all integer, and is thus optimal for the ILP, then  $X^0$  is optimal IQP, and  $g(X^0, W^0) = f(X^0)$ .

Note that the two objective function values  $g(X^0, W^0)$  and  $f(X^0)$  need not be equal if some of the variables  $x(i,k)$  have noninteger values.

Of course, we shall not often be lucky enough to be able to apply F3, since the optimal solution to the RILP will not generally be all integer. The next result provides a means for establishing an all-integer solution that improves on a partially fractional solution to the RILP. In particular, we exploit the fact that, in the objective function of the IQP, no terms raise any individual  $x(i,k)$  to the second power; i.e., all the quadratic terms are cross products. By fixing the  $x(i,k)$  for all  $i$  not equal to some  $i^*$ , the IQP's objective function becomes linear in the remaining  $x(i,k)$  variables. We exploit this observation.

*Proposition.* If  $(X^0, W^0)$  is feasible for the RILP, then there is a feasible solution  $X^1$  for the IQP satisfying  $f(X^1) \leq f(X^0)$ .

*Proof.* If  $X^0$  is integer, then by F1,  $X^1 = X^0$  suffices. Therefore, assume that  $X^0$  has at least one fractional value, say  $x^0(i,k')$ . Consider the subvector  $X(i)$  of  $X$  defined by

$$X(i) = \{ x(i,k) \mid k \in S(i) \}.$$

Suppose that we allow the variables of  $X(i)$  to vary while requiring all other  $X$  variables, including all the  $x(k,k)$ , to retain their values in  $X^0$ . By substituting the fixed values,  $\{x^0(j,k); j \neq i; k \in S(j)\}$ , into the IQP, and relaxing the integrality constraints, we obtain the following linear program in  $X(i)$ :

Minimize

$$\sum_{k \in S(i)} \gamma(i,k) x(i,k) + K$$

subject to

$$\sum_{k \in S(i)} x(i,k) = 1$$

$$x(i,k) \leq x^0(k,k) \quad \text{for } k \in S(i)$$

$$x(i,k) \geq 0 \quad \text{for } k \in S(i)$$

where

$$\gamma(i,k) = v(i) c(i,k) - (2b-d) \sum_{j=1}^N [v(i,j) + v(j,i)] x^0(j,k),$$

and  $K$  is a constant specified by

$$K = \sum_{k=1}^m a(k) x^0(k,k) + \sum_{j=1}^N v(j) \sum_{k \in S(j)} c(j,k) x^0(j,k) \\ - (2b-d) \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\substack{q=1 \\ q \neq i}}^N v(j,q) \sum_{k \in S(j,q)} x^0(j,k) x^0(q,k).$$

Recall that, since  $(X^0, W^0)$  is feasible for the RILP, each  $x^0(k,k)$  has value zero or one.

We determine the optimal solution  $X'(i)$  to this linear program by finding

$$\gamma'(i) = \min_k \{\gamma(i,k)\}.$$

Assume that  $k = k'$  solves this problem, and thus

$$\gamma'(i) = \gamma(i,k').$$

An optimal vector  $X'(i)$  is given by

$$x'(i,k') = 1$$

$$x'(i,k) = 0 \text{ for } k \neq k'.$$

Furthermore,

$$\gamma'(i) = \gamma(i,k') = \sum_{k \in S(i)} \gamma(i,k) x'(i,k) \leq \sum_{k \in S(i)} \gamma(i,k) * x^0(i,k).$$

Let  $X^1$  be a vector defined by

$$X^1(i) = X'(i),$$

$$X^1(j) = X^0(j) \text{ for } j \neq i.$$

Note that  $X^1$  is feasible for the RILP, that it has fewer fractional values than  $X^0$  because the  $x^1(i,k)$  are now integer for all  $k \in S(i)$ , and that

$$f(X^0) = \sum_{k \in S(i)} \gamma(i,k) * x^0(i,k) + K,$$

$$f(X^1) = \sum_{k \in S(i)} \gamma(i,k) * x^1(i,k) + K.$$

Therefore,

$$f(X^1) \leq f(X^0).$$

Moreover,  $X^1$  is a feasible solution for the IQP with integrality constraints relaxed.

Now if  $X^1$  is integer, which means that the  $x(j,k)$  are integer for all  $j \neq i$ , then we have established the desired result. Otherwise, we repeat the argument with some index  $j$  for which  $x(j,k)$  is fractional for some  $k$ . Eventually, we exhaust all such indices and find the desired integer solution  $X^1$ .

Since the proposition applies to every feasible solution to the RILP, it also applies to the RILP's optimal solution. Note that we do not claim that the  $X^1$  found from the RILP's optimal is optimal for the IQP. Nor have we claimed that the transformation from  $X^0$  to  $X^1$  preserves objective function orderings in the sense that

$$g(X^1, W^1) \leq g(X^0, W^0) \text{ implies that } f(X^1) \leq f(X^0).$$

The proposition does show, however, that solving the RILP, with far fewer integer variables than the ILP, combined with the transformation  $X^0 \rightarrow X^1$ , provides a feasible solution and, hence, a valid upper bound to the optimal objective value of the IQP and ILP. This feasible solution might conceivably be a very good final solution or provide a useful upper bound for a branch and bound procedure. Moreover, this possibility helps to justify our decision to branch on the  $x(k,k)$  variables before the  $x(i,k)$  variables in our branch and bound procedure.

Finally, we note that the proposition remains valid whenever  $f(\cdot)$  is concave in  $X(i)$  for any  $i$ , with fixed  $X(j)$  for all  $j \neq i$ . In addition, the proposition will apply for a nonconcave  $f(\cdot)$  whenever the NLP in  $X(i)$ , without integrality constraints, always has an integer solution. The proof is similar.

## 4. BRANCH AND BOUND SOLUTION METHOD.

### 4.1 DEVELOPMENT AND BACKGROUND.

The linear program based branch and bound technique uses model specific selection rules that exploit the model's structure.

To avoid confusion with the term "node", we use "vertex" to refer to a node in the branch and bound tree and "city" to denote a node in the original problem. A vertex represents a solution to the linear program with one or more variables set to an integer value. Each vertex contains both locked and unlocked integer variables. In the branch and bound tree, a locked variable has been set to its integer value by a branch along the path from the root to a vertex.

Every branch and bound procedure requires two types of rules: one for vertex selection and one for branch selection. Some recent literature (Salkin [37], Akinc and Khumwala [1], Crowder, Johnson and Padberg [10], and Johnson, Kostreva and Suhl [20]) has used the following choice for these selections.

1. Vertex selection.
  - a) best objective function value,
  - b) fewest fractional values,
  - c) fewest unlocked variables, and
  - d) a hybrid of (a), (b) and (c).



## 2. Branch selection.

- a) fractional variable closest to 1,
- b) fractional variable closest to 0, and
- c) fractional variable that is next in line according to an ordering scheme specific to the problem.

Akinc and Khumawala [1], in discussing a capacitated warehouse location problem formulated as a mixed integer program, point out that the vertex selection rule (a) may require large amounts of computer storage but small amounts of computer time, while the vertex selection rules (b) and (c) require the opposite. This time versus storage dilemma led them to employ a hybrid rule, which uses the vertex selection rule (a) until the number of nonterminal vertices reaches a user-specified level, and then switches to rule (c) to clean up some of the nonterminal vertices. Upon doing so, they reinstate rule (a).

Their branch selection rule is interactive and of the type (c). It re-orders the integer variables at a vertex prior to branching from it, by solving auxiliary linear programs to approximate the effect of changing the open versus closed status of any warehouse.

Their conclusions concerning computer time and storage depend in part upon other factors specific to the problem, such as the warehouse capacity constraints, which they relax during the branching process.

Johnson et al. [20], like Akinc and Khumawala, tried various vertex selection rules for a set of project selection planning models and found the hybrid the most effective. For branch selection, they order the variables by first classifying them into three categories: project variables, activity variables, and fixed charge variables. They report that they never obtained good results on larger problems without forcing the fixed charge variables to be selected at the top of the tree. If set to zero, these variables force many other variables to be zero.

Our experience with several branch and bound strategies for the earth station location model supports many of the findings of these authors. We used the linear programming package XMP, developed by Marsten [30]. Prior to writing a branch and bound program to call XMP as a subroutine, we attempted to solve problems of networks of moderate size ( $N \leq 25$ ,  $m \leq 18$ ) by performing the branch and bound interactively after each execution of the linear program.

We employed vertex selection rule (a) and occasionally rule (d).

We restricted our branching variables to the  $x$  variables alone because, as shown in Section 2.5, the  $w$  variables will be forced to be integer whenever all the  $x$  variables are integer. This result is fortuitous because there are many more  $w$  variables than  $x$  variables. Within this limitation, we attempted to employ branch selection rules (a) and (b) as well as a rule of "fractional variable furthest from integer".

In every problem in which we attempted this strategy, we encountered one of two outcomes: (i) a tree of more than 50 vertices, none of which were close to being all integer, and (ii) a tree of many vertices containing all fractional variables equal to one half. Either outcome foiled our branching selection rules. We then decided to use the branching rule of " $x(i,k)$  variable corresponding to highest demand city  $i$ ". Unlike its predecessor, this rule is unlikely to produce ties. More importantly, it considers first those location decisions most likely to have greatest impact on the overall cost.

The final developmental step was a two-phase branching strategy similar to the categorization of variables employed by Johnson, et al. We divided the  $x$  variables into two categories:  $x(k,k)$  (earth station location), and  $x(i,k)$  for  $i \neq k$  (terrestrial connection).

These two factors -- the categorizing of the variables and the ordering of cities by size of demand -- greatly improved the branch and bound efficiency. Section 5 presents our computational results.

## 4.2 THE PROBLEM-SPECIFIC BRANCH AND BOUND METHOD

The problem-specific branch and bound technique is implemented in a FORTRAN program called LOCAT, which calls XMP as a subroutine. The technique developed in an attempt to trim the branch and bound tree as quickly as possible. We chose the two phase strategy not only for the reasons elucidated by Johnson et al., but also because constraint (3) of the model forces the  $x(i,k)$ 's to be fractional whenever the  $x(k,k)$ 's are fractional, but not vice versa.

Phase 1 first chooses cities with large circuit requirements as branching variables in order to determine early whether the communication network is less costly with or without an earth station at that city. The size of circuit demand remains the criterion for selection in phase 2 in order to determine as quickly as possible the best homing bases for these non-earth station cities. By employing the size criterion, we avoid fixing a high impact city while deep in the branch and bound tree.

### 4.2.1 *Vertex Selection*

The algorithm's selection rule gives priority to vertices with fractional  $x(k,k)$ : a vertex with all integer  $x(k,k)$  is ineligible for propagation until every non-terminal vertex contains all integer  $x(k,k)$ . (A terminal vertex has objective function value exceeding the lower bound.) More specifically, the vertex selection rules can be described as follows.

1. If the tree contains one or more vertices with fractional  $x(k,k)$ , then choose, from among those vertices, the one possessing the lowest objective function value.
2. If the tree contains no vertices with fractional  $x(k,k)$ , but one or more vertices with fractional  $x(i,k)$ , then branch from the vertex with the lowest objective function value.

When all non-terminal vertices have all integer  $x(i,k)$ , then the branch and bound tree has been completed, and the optimal configuration is the least cost solution from among those vertices with all integer solutions.

#### 4.2.2 *Branch Selection*

We used the following branch selection rules. Note that the second rule exploits the multiple choice effect produced by constraints (2) of the model.

1. If the solution at vertex  $j$  has one or more fractional  $x(k,k)$ , then choose the fractional  $x(k,k)$  corresponding to the city  $k = k'$  with the largest total circuit requirements. To create actual branches, form two new vertices by setting the variable  $x(k',k')$  first to zero and then to one.
2. If vertex  $j$  has no fractional  $x(k,k)$ 's, but has one or more fractional  $x(i,k)$ 's, then choose, from among the set of cities  $i$  for which there is a fractional  $x(i,k)$ , that city  $i = i'$  with the largest circuit requirements. Then, for each city in the set  $\{k \in S(i') \mid x(k,k) \text{ not locked at zero in vertex } j\}$ , create a branch from vertex  $j$  by setting  $x(i',k) = 1$ .

#### 4.2.3 *Example*

Figure 4 presents an actual branch and bound tree created by this strategy. The first upper bound results from a user-given configuration. The branch and bound process itself creates the second upper bound at vertex 23. The second phase begins after vertex 22, with five branches created at vertex 6.

homing stations, but run three has five additional cities. Yet the smaller network requires more pivots and more branch and bound vertices. An important factor is that the network for run three is sparser than that for run two. Its sixteen candidate homing stations are spread over a larger area than are the sixteen candidates for run three. Thus, the distance parameter ( $r=500$ ) forces the candidate sets  $S(i)$  to be smaller for the more dispersed network. Consequently, run three has fewer integer variables.

All runs were executed on an IBM 3081. Accurate computer CPU times are not available, but no run exceeded thirty minutes.

## 5.2 COMPARISONS WITH OTHER METHODS

Optimization methods are useful not only in generating "good" solutions but also in providing benchmarks against which to compare other methods. Moreover, the development of any optimization procedure poses a natural question: "Does the method find solutions with significant cost savings over solutions found by simpler methods?" To answer this question and test whether a heuristic method might prove to be successful in this problem setting, we compared the optimal configuration against configurations found by three other methods: an educated eyeball procedure and two types of greedy heuristic, a demand dependent greedy procedure and a cost dependent greedy procedure. The educated eyeball solution and the demand dependent greedy had already generated solutions under consideration by management. Our comparisons were made on a twenty-five city network. To accommodate uncertainty in the cost data, we made these comparisons for several different sets of cost parameters, four of which are depicted in Table III. We used the same demand data in all runs.

Some description of the two heuristics is needed.

## 5. COMPUTATIONAL EXPERIENCE

This section summarizes computational experience in applying the algorithm to actual networks ranging in size from ten to forty cities and to problems with a variety of cost parameters. It also introduces two heuristic methods for solving the problem and describes computational experience in applying these heuristics.

### 5.1 PROBLEM SIZE

Table II summarizes problem features for networks varying in size from twelve nodes with eight candidate homing stations to forty nodes with twenty candidates. The circuit requirements for these examples represent actual historical telecommunication usage, and the cost parameters represent current telecommunication earth station capital costs as well as terrestrial leasing cost. The demand nodes represent a hypothetical collection of major United States cities that reflect both geographical dispersion and variation in population. The same cost parameters and circuit requirements were used throughout all five runs. A distance parameter  $r$  was used to limit the candidate set  $S(i)$  of each node  $i$  to those candidate homing stations within  $r$  miles from node  $i$ .

As illustrated in Table II, in the linear program tableau, the number  $C$  of constraints far exceeds the number  $V$  of variables. Accordingly, in each case, the model solved the dual linear program at each vertex of the branch and bound tree. Had we employed the more straightforward  $y(i,j,k)$  variable replacement, then run 5 would have required 548  $y$  variables, as opposed to the 251  $w$  variables, and thus a dual matrix with 688, as opposed to 391, dual constraints.

It is important to note that increasing the number of cities and candidate homing stations does not necessarily add greatly to computation time. The relative density of the network is also a factor. For example, runs two and three have the same number of candidate

*DEMAND DEPENDENT GREEDY HEURISTIC*

- i) Place homing stations on the two highest demand cities not less than  $\mu$  miles apart, where  $\mu$  is a user defined parameter.
- ii) Home each of the remaining  $N-2$  cities into its closest homing station city and calculate the cost of this configuration.
- iii) Locate a homing station at the third highest demand city that is no closer than  $\mu$  miles from the first two earth stations, and rehome any of the remaining  $N-3$  cities that are closest to the third earth station.
- iv) Continue in this manner until one of two events occur: either the addition of no single homing station decreases the configuration cost, or each nonhoming station city is within  $\mu$  miles of a homing station.

For the twenty-five city example, with  $\mu = 500$ , this procedure produced a maximum of six homing stations, while for  $\mu = 300$  it produced a maximum of nine. In the latter case, the heuristic did not terminate until all candidate earth station sites were within 300 miles of a chosen earth station site.

*COST DEPENDENT GREEDY HEURISTIC*

- i) Among all solutions with two earth stations, in which each non-homing station city attaches to its closest earth station, choose the one with minimum cost. Set  $k=2$  and  $C^k$  equal to the cost of this solution.
- ii) Find the earth station location  $q$  whose addition to the current solution gives the least cost  $C^{k+1}$  when every non-homing station attaches to its closest earth station.

- iii) If  $C^{k+1} \geq C^k$ , terminate with the  $k$  earth station solution producing  $C^k$ . Otherwise, set  $k$  equal to  $k + 1$  and proceed to step (ii).

Table III compares these four methods (optimal, two heuristics, and educated eyeball), each with cost parameters  $a = \$40,000$ ,  $c = \$1$ , and  $d = 0$ , and four different sets of cost parameters  $b$ , for a problem of twenty-five cities and eighteen candidate homing locations. For each problem, each set  $S(i)$  is defined by a radius of 400 miles, that is,  $k \in S(i)$  if and only if  $i$  is not more than 400 miles from  $k$ . For each set of cost parameters, the demand dependent greedy algorithm improved over the educated eyeball approach, and the optimal solution improved over the demand dependent greedy solution by from 2.5% to 33%. In two of the four cases, the cost dependent greedy heuristic produced the optimal solution. For the case  $b = \$300$ , this heuristic produced a solution with one additional earth station and a cost 3.3% higher than the optimal solution. In another case ( $b = \$150$ ), the cost dependent greedy heuristic found the optimal placement of earth stations but not the optimal homing assignments, and a cost within one percent of the optimal. Because the cost dependent greedy heuristic assigns each city to its closest earth station, it does not necessarily find the optimal clustering of nodes. Thus, it will miss the optimal homing assignment whenever the optimal solution contains high demand, non-earth station city pairs "near" each other but sufficiently distant that a different earth station is closest to each.

Figures 5, 6, 7 depict configurations found by the four different methods, using cost parameters  $a = \$40,000$ ,  $b = \$150$ ,  $c = \$1$ ,  $d = 0$ . Figure 5 represents a management plan of nine homing stations. Figure 6 represents an alternative management plan, found by the demand dependent greedy algorithm, also of nine homing stations, but not the same nine, and at a cost of one million dollars less. Figure 7 shows the optimal solution of 17 earth stations, giving a cost reduction of 16 percent over the first management plan and 8 percent over the demand dependent greedy algorithm solution. The dashed line in Figure 7 shows the homing assignment of the cost dependent greedy solution.

In experiments with other practical choices of the cost parameters (e.g.,  $a = \$60,000$  and values of  $b$  between  $\$40$  and  $\$300$ ), the cost dependent greedy heuristic always found



the optimal placement of earth stations. Moreover, when we used the demand dependent greedy solution as a starting configuration for the cost dependent greedy heuristic, the procedure also gave the optimal solution, except for the case of  $a = \$40,000$  and  $b = \$300$ . In this case, the two phased demand dependent/cost dependent greedy heuristic produced a solution with value \$25.56 million - a slight improvement over either greedy procedure applied by itself.

Our computational results suggest that, for a wide variety of realistic cost parameters, the optimal solution improves significantly on the costs of both an educated eyeball plan and a demand dependent greedy heuristic, but that the cost dependent greedy performs almost as well as optimization.

## 6. POTENTIAL MODELING AND ALGORITHMIC ENHANCEMENTS

The model considered in this paper addresses a basic combinatorial feature of satellite network applications--namely, the effect of grouping demand nodes into local clusters whose members all communicate with each other via a local switch and communicate with members of other clusters via a satellite. There are several potential avenues for fruitfully enriching the basic model:

- i) permitting multiple routes for transmitting traffic between pairs of nodes,
- ii) permitting minor nodes to home into more than one earth station,
- iii) limiting the capacity of the earth stations or the satellite(s),
- iv) permitting earth stations to communicate with each other directly via a terrestrial link, and/or
- v) permitting the system to use several different types of earth stations, as is customary in practice; the model must then account for the fact that not all earth stations are compatible (some of them cannot communicate directly with others).

These enhancements would permit the model to account for important reliability issues as well as to incorporate important technological features of communications systems (e.g., limited capacities, multiple technologies).

Additional algorithmic enhancements might be possible as well. For example, rather than introducing the variables  $w$  as defined in (7), we might incorporate the term

$$1 - \max_{k \in S(i,j)} \{|x(i,k) - x(j,k)|\} \quad (12)$$

in place of  $w(i,j)$  in the objective functions of the integer linear program ILP. Note that since the term  $|x(i,k) - x(j,k)|$  is piecewise linear as a function of  $x(i,k)$  and  $x(j,k)$ , and since the maximum of a finite number of piecewise linear and convex functions is also piecewise linear and convex, the entire term (12), because of the negative sign before the max term, is piecewise linear and concave as a function of  $X$ . Consequently, replacing the  $w(i,j)$  variables in the objective function (10) by the term (12), and noting that the coefficient of each  $w$  in (10) is negative, we see that the objective function is piecewise linear and convex as a function of  $X$ . Consequently, instead of introducing the  $w$  variables and the large number of constraints (8) and (9), we could instead solve the linear programming relaxation of MILP with only the  $X$  variables by using a piecewise linear programming implementation of the simplex method (e.g., see Fourer [14], and Premoli [35]). Our computational experience indicates that for the size and range of problems that we have been considering, this algorithmic refinement is not necessary. Conceivably, however, for larger size problems or for problems that incorporate some of the modeling enhancements that we have mentioned in this section, the refined algorithm might prove to be very useful.

## 7. SUMMARY

This paper was motivated by the flexibility that satellite transmission technologies offer in communication systems as a complement to traditional terrestrial transmission. We have modeled the topological design of a voice carrying satellite communication network as a quadratic facility location problem. The model contains two types of decision variables, modeling the location of homing stations and the assignment of customers (cities) to homing stations. The objective function is quadratic because the local switching costs for terrestrial communication between customers assigned to the same homing station differ from the costs necessary for communication via satellite between customers assigned to different homing stations.

We have formulated a zero-one quadratic programming model of the earth station location problem and recast it as an equivalent linear mixed integer program. For a typical application context with forty demand nodes (cities), the quadratic facility location model contains approximately 140 zero-one variables. A standard linearization of this problem would require approximately 550 additional zero-one variables. However, in this problem the additional variables can be modeled as continuous. Furthermore, by exploiting the problem's special structure, we can reduce the required additional variables to 250 and we can use a specialized branch and bound procedure whose number of vertices has never exceeded one fifth of the number of integer variables, even for fairly dense networks.

The applications of this modeling approach to real data shows that optimization can improve considerably on managerially generated solutions. For one 25 node example, the optimization model locates twice as many earth stations as did both a demand dependent greedy heuristic and an eyeball approach. Applications of a wide variety of cost parameters suggest the optimization model can provide cost savings of up to 33%.

We have also shown that a cost dependent greedy solution provides an optimal placement of earth stations in all cases that we tested. This algorithm has the advantage of being easy to program; it also is very fast and requires little storage and, therefore, can

be implemented conveniently even by on a microcomputer (our implementation used BASIC on an IBM PC). The optimization approach, on the other hand, has the advantage of easily incorporating a wide variety of additional constraints and problem features. Whether the cost dependent greedy heuristic, or some appropriate modification of it, will work well in these situations requires further testing. Our results show that this type of testing might be attractive to pursue and even if the heuristic does not continue to perform well in more complex settings, a special purpose branch and bound approach is likely to be very effective.

## REFERENCES

1. U. Akinc and B. M. Khumawala, An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Sci.* 23 (1977) 585-594.
2. A. Assad, Models for rail transportation. *Transportation Res.* 14A (1980) 205-220.
3. A. Balakrishnan and S.C. Graves, A composite algorithm for the concave-cost LTL consolidation problem. WP#1669-85, Sloan School of Management, M.I.T., Cambridge, MA (1985).
4. M. Balinski and P. Wolfe, On Benders decomposition and a plant location problem. ARO-27, *Mathematica*, Princeton, NJ (1963).
5. O. Bilde and J. Krarup, Sharp lower bound and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics* 1 (1977) 78-97.
6. L. Bodin, B. Golden, A. Schuster and W. Romig, A model for the blocking of trains, *Transportation Res.* 14B (1980) 115-120.
7. R. R. Boorstyn and H. Frank, Large-scale network topological optimization. *IEEE Transactions on Communications* COM-25 (1977) 29-47.
8. N. Christofides and J. E. Beasley, Extensions to a Lagrangian relaxation approach for the capacitated warehouse location problem, *European Journal of Operational Research* 12 (1983) 19-28.
9. G. Cornuejols, M. L. Fisher and G. L. Nemhauser, Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Sci.* 23 (1977) 789-810.

10. H. Crowder, E. L. Johnson and M. W. Padberg, Solving large-scale zero-one linear programming problems. *Operations Res.* 31 (1983) 803-834.
11. P. S. Davis and T. L. Ray, A branch and bound algorithm for the capacitated facility location problem. *Naval Res. Logistics Quarterly* 16 (1969) 331-344.
12. M. A. Effroymsen and T. L. Ray, A branch and bound algorithm for plant location. *Operations Res.* 14 (1966) 361-369.
13. D. Erlenkotter, A dual-based procedure for uncapacitated facility location. *Operations Res.* 26 (1978) 992-1009.
14. R. Fourer, A simplex algorithm for piecewise-linear programming I: derivation and proof. *Math. Programming* 33 (1985) 204-233.
15. R. Francis and P. B. Mirchandani, *Discrete Location Theory*, John Wiley and Sons, New York (forthcoming).
16. R. Francis and J. White, *Facility Layout and Locations: An Analytic Approach*. Prentice-Hall, Englewood Cliffs, NJ (1974).
17. B. Gavish, Topological design of centralized computer networks -- formulations and algorithms. *Networks* 12 (1982) 355-377.
18. M. Guignard and K. Spielberg, A direct dual method for the mixed plant location problem with some side constraints. *Math. Programming* 17 (1979) 198-228.
19. G. Y. Handler and P. B. Mirchandani, *Location on Networks: Theory and Algorithms*. M.I.T. Press, Cambridge, MA (1979).

20. E. L. Johnson, M. M. Kostreva and U. H. Suhl, Solving 0-1 integer programming problems arising from large scale planning models. *Operations Res.* 33 (1985) 803-820.
21. A Kershenbaum and R. R. Boorstyn, Centralized teleprocessing network design. *Networks* 13 (1983) 279-293.
22. B. M. Khumawala, An efficient branch and bound algorithm for the warehouse location problem. *Management Sci.* 18 (1972) 718-731.
23. A. A. Kuehn and M. J. Hamburger, A heuristic program for locating warehouses. *Management Sci.* 9 (1963) 643-666.
24. B. W. Lamar, Y. Sheffi and W. B. Powell, Bounding procedures for fixed charge, multicommodity network design problems. Working Paper, Dept. of Civil Engineering, M.I.T., Cambridge, MA (1984).
25. J. M. Y. Leung, Polyhedral structure of capacited fixed charge problems and a problem in delivery route planning, Ph. D. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA (1985).
26. J. M. Y. Leung, T. L. Magnanti and V. Singhal, Point-to-point delivery systems, *ORSA/TIMS Bulletin* (November 1984) page 42.
27. H. Luss, Operations research and capacity expansion problems: a survey. *Operations Res.* 30 (1982) 907-947.
28. T. L. Magnanti and R. T. Wong, Network design and transportation planning: models and algorithms. *Transportation Sci.* 18 (1984) 1-55.



29. A. S. Manne, Plant location under economies-of-scale -- decentralization and computation. *Management Sci.* 11 (1964) 213-235.
30. R. E. Marsten, The design of the XMP linear programming library, *ACM Transactions on Mathematical Software* 7 (1981) 481-497.
31. R. Mendelsohn, R. R. Boorstyn and A. Kershenbaum, A cutting plane approach to the concentrator location problem. Working Paper (1984).
32. A. Mirzaian, Lagrangian relaxation for the star-star concentrator location problem: approximation algorithm and bounds. *Networks* 15 (1985) 1-20.
33. R. M. Nauss, An improved algorithm for the capacitated facility location problem. *Journal of the Operational Research Society* 29 (1978) 1195-1201.
34. W. B. Powell and Y. Sheffi, The load planning problem of motor carriers: problem description and a proposed solution approach. *Transportation Res.* 17A (1983) 471-480.
35. A. Premoli, Piecewise-linear programming: the compact (CPLP) algorithm. *Mathematical Programming* 36 (1986) 210-227.
36. G. Sa, Branch and bound and approximate solutions to the capacitated plant location problem. *Operations Res.* 17 (1969) 1005-1016.
37. H. M. Salkin, *Integer Programming*. Addison-Wesley, Reading, MA (1975).
38. V. Singhal, Point-to-point package delivery systems. M.Sc. Thesis, Operations Research Center, M.I.T., Cambridge, MA (1984).

39. K. Spielberg, Algorithms for the simple plant-location problem with some side conditions. *Operations Res.* 17 (1969) 85-111.
40. J. F. Stollsteimier, A working model for plant numbers and locations. *Journal of Farm Economics* 45 (1961) 631-645.
41. A. S. Tanenbaum, *Computer Networks*. Prentice-Hall, Englewood Cliffs, NJ (1981).
42. B. C. Tansel, R. L. Francis and T. J. Lowe, Location on networks: a survey, parts I and II. *Management Sci.* 29 (1983) 482-511.
43. R. T. Wong, Location and network design. In *Combinatorial Optimization: Annotated Bibliographies* (M. O'hEigearthaigh, J. K. Lenstra and A. H. G. Rinnooy Kan, eds.) Wiley-Interscience, New York (1985) pp. 127-147.

**Table I**  
**Comparison of linearized and true values of clustering variables**

	x VARIABLES						RESULTING RHS OF CONSTRAINTS				RESULTING w(i,j) VALUES		TRUE
	(i,k)	(j,k)	(i,l)	(j,l)	(i,p)	(j,p)	(8-k)	(8-l)	(8-p)	(9)	(8), (9)	(8), (9), (10)	w(i,j)
<b>Case 1</b>	1	1	0	0	0	0	≤ 1	≤ 1	≤ 1	≤ 2	≤ 1	1	1
<b>Case 2a</b>	1	0	0	1	0	0	≤ 0	≤ 0	≤ 1	≤ 2	0	0	0
<b>Case 2b</b>	1	0	0	1	0	0	≤ 0	—	≤ 1	≤ 1	0	0	0
<b>Case 2c</b>	1	0	0	1	0	0	—	—	≤ 1	≤ 0	0	0	0

Table II  
Network size, problem size, and computational time

RUN	N	m	M	M/N	C	V	PIVOTS IN INITIAL LP		VERTICES IN B&B TREE
							TO BASIC FEASIBLE SOLUTION	TO OPT. SOLUTION	
1	12	8	73	6.1	762	138	124	957	12
2	25	16	117	4.7	1224	255	186	1784	14
3	30	16	95	3.2	937	235	217	1146	8
4	40	16	114	2.8	1301	346	217	1148	4
5	40	20	140	3.5	1508	391	279	1109	27

#### PARAMETERS THROUGHOUT

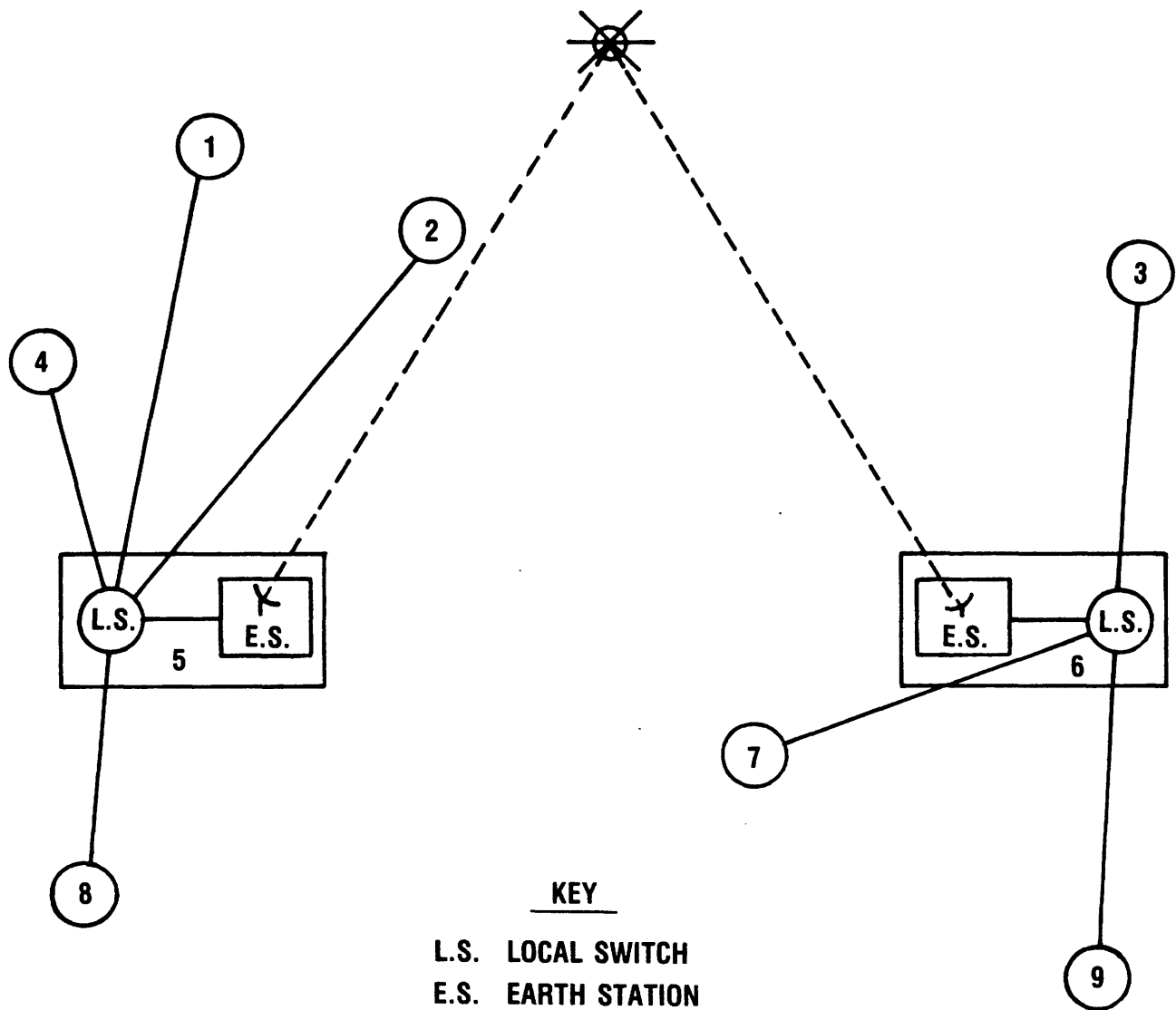
- a = \$40,000 = earth station startup cost
- b = \$300 = earth station per circuit cost
- c = \$1 = terrestrial link per circuit mile cost
- d = \$0 = local switch per circuit cost
- r = 500 miles = max allowable distance for node i to home to node k

#### NOTATION

- N = number of cities
- m = number of candidate homing stations
- M = number of x variables in primal
- C = number of primal constraints (dual structural variables, no slacks)
- V = number of primal structural variables, no slacks (dual constraints)
- M/N = network density factor

**TABLE III**  
**Cost Comparison for a 25 City Network**

	a = 40 K, b = 40, c = 1, d = 0		a = 40 K, b = 80, c = 1, d = 0		a = 40 K, b = 150, c = 1, d = 0		a = 40 K, b = 300, c = 1, d = 0	
	Cost	No. of ES's	Cost	No. of ES's	Cost	No. of ES's	Cost	No. of ES's
<b>Educated Eyeball</b>	\$10.93 M	9	\$13.44 M	9	\$17.84 M	9	\$27.27 M	9
<b>Demand Dependent Greedy</b>	\$ 9.15 M	9	\$11.73 M	9	\$16.24 M	9	\$25.71 M	7
<b>Cost Dependent Greedy</b>	\$ 6.09 M	18	\$ 9.43 M	18	\$15.05 M	17	\$25.91 M	10
<b>Optimal</b>	\$ 6.09 M	18	\$ 9.43 M	18	\$14.96 M	17	\$25.08 M	9
<b>Savings: Optimal vs Eyeball</b>	\$ 4.87 M	44%	\$ 4.01 M	30%	\$ 2.88 M	16%	\$ 2.19 M	8.0%
<b>Savings: Optimal vs Demand Dependent Greedy</b>	\$ 3.06 M	33%	\$ 2.30 M	20%	\$ 1.28 M	7.9%	\$ .63 M	2.5%
<b>Savings: Optimal vs Cost Dependent Greedy</b>	0	0%	0	0%	0	0.7%	\$ .83 M	3.3%



**NODES 5 and 6 ARE HOMING STATIONS**  
**NODES 1,2,3,4,7,8,9 ARE NOT**

Figure 1. Example of a complete satellite network topological design.

# LEAST COST MULTI-MEDIA COMMUNICATION NETWORK MODEL

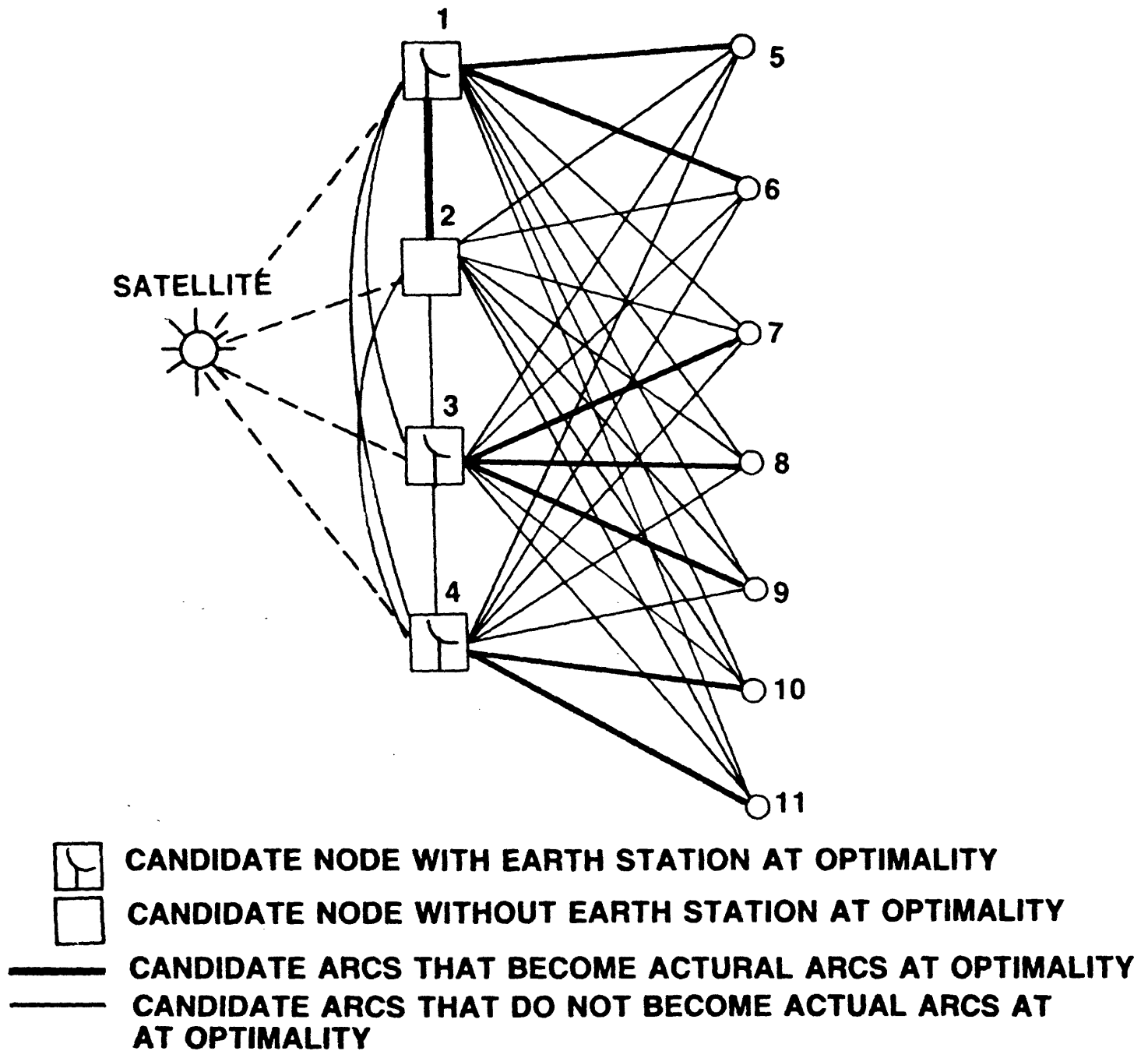


Figure 2. Transshipment network formulation with possible optimal solution.

...  $x(3,1)$   $x(4,1)$   $x(5,1)$   $x(3,2)$   $x(4,2)$   $x(5,2)$  ...

⋮	⋮	○			○			⋮
$x(3,1)$		0	$u(3,4)$	$u(3,5)$	0	0	0	
$x(4,1)$	○	$u(3,4)$	0	$u(4,5)$	0	0	0	○
$x(5,1)$		$u(3,5)$	$u(4,5)$	0	0	0	0	
$x(3,2)$	○	0	0	0	0	$u(3,4)$	$u(3,5)$	○
$x(4,2)$		0	0	0	$u(3,4)$	0	$u(4,5)$	
$x(5,2)$		0	0	0	$u(3,5)$	$u(4,5)$	0	
⋮	⋮	○			○			⋮

**Note**

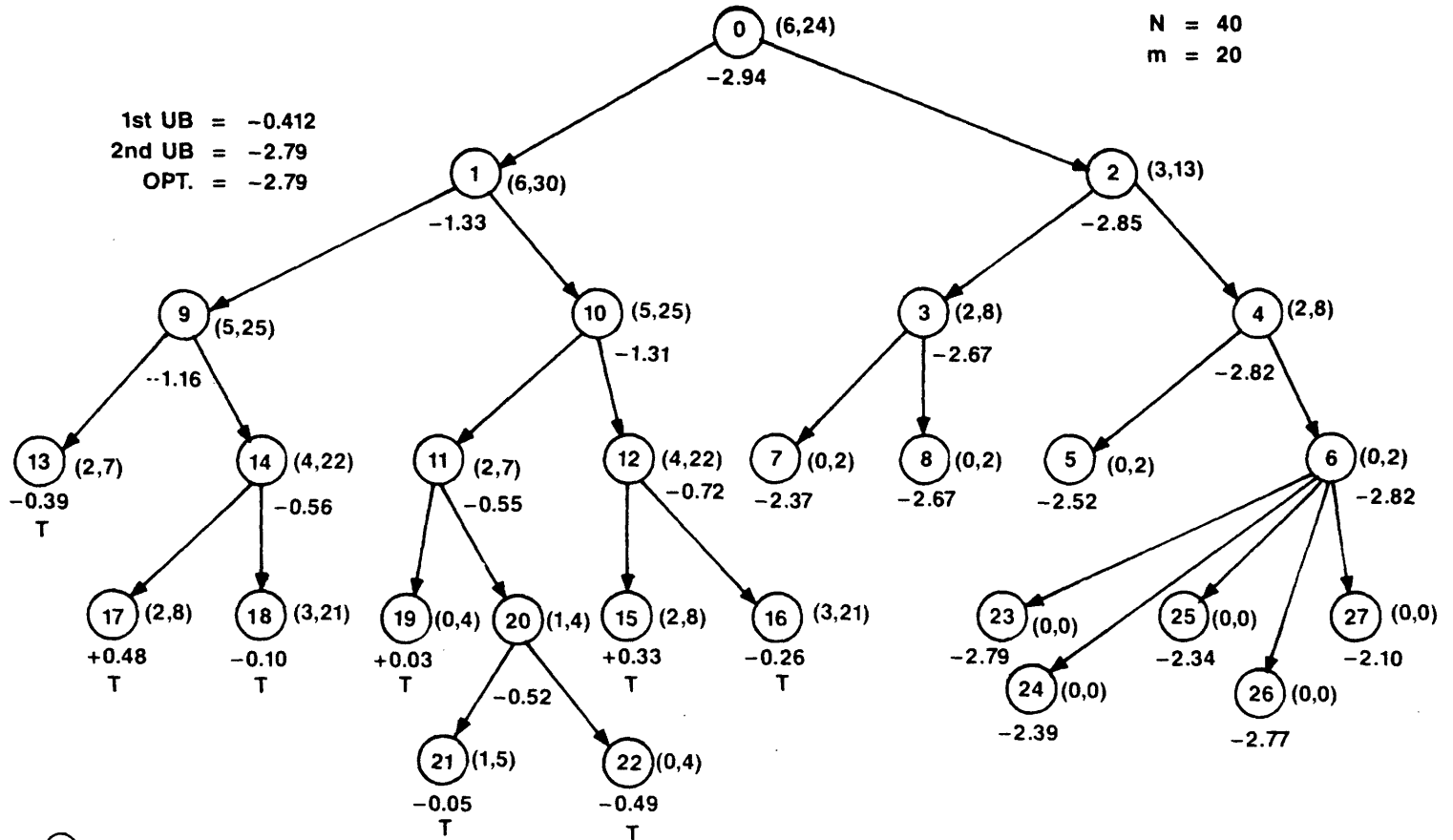
$$u(i,j) = v(i,j)/2$$

In this example,  $S(3) = S(4) = S(5) = \{1,2\}$

Figure 3. Example of matrix of coefficients of quadratic terms.



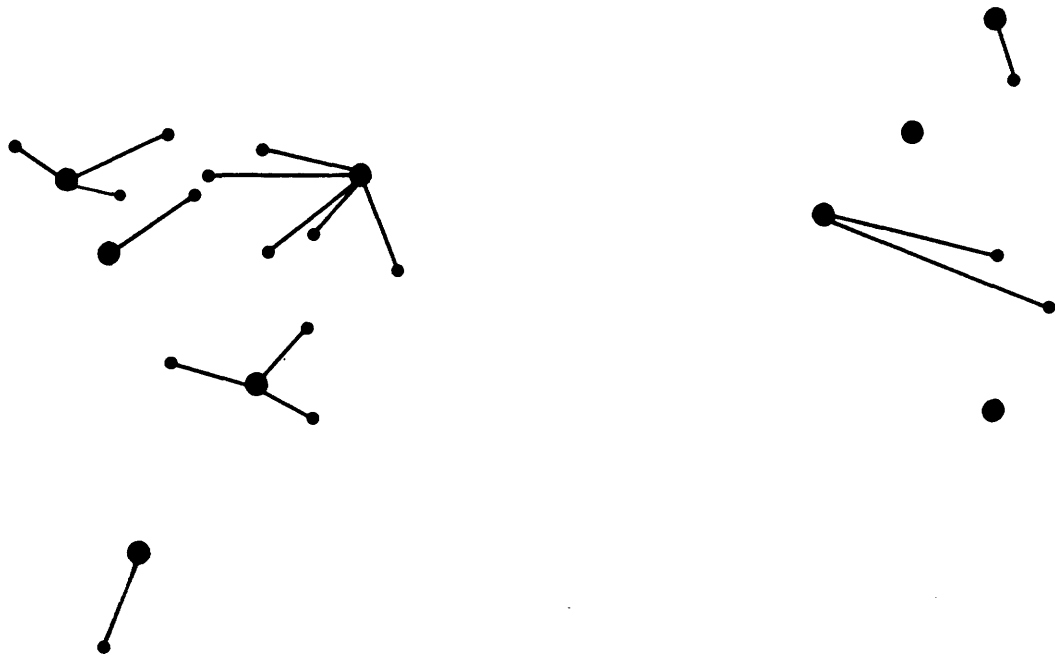
# BRANCH AND BOUND TREE



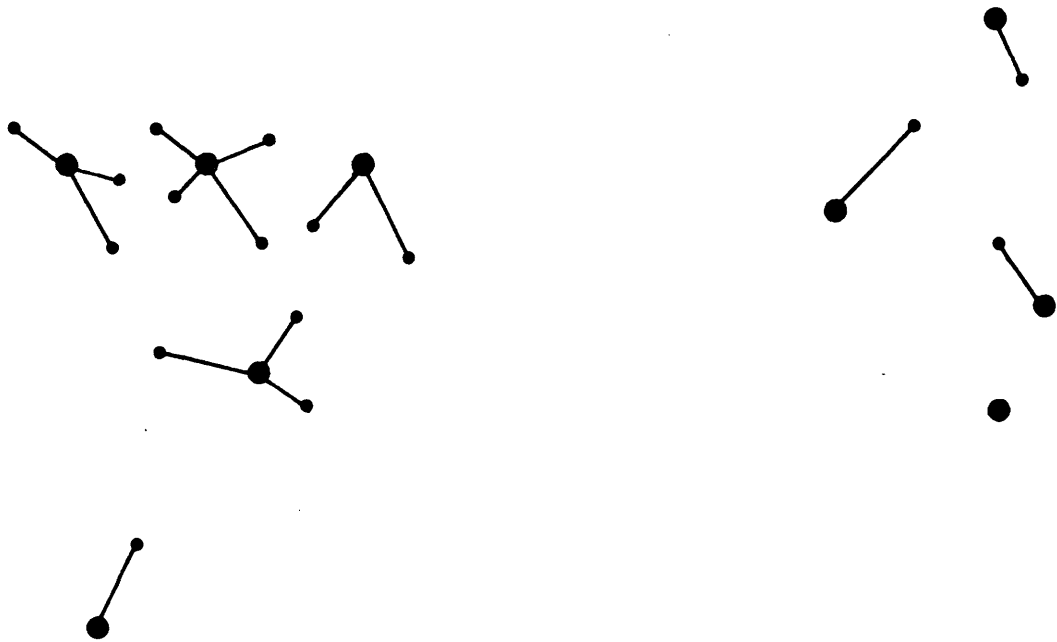
- Ⓛ (p,q)
- L = VERTEX NO.
- p = NO. OF FRACTIONAL  $x_{kk}$
- q = NO. OF FRACTIONAL  $x_{ik}$
- .xxx = OBJ. FUNCT. VALUE IN \$MILLIONS
- T DENOTES TERMINAL VERTEX

NOTE: SOME OBJECTIVE FUNCTION VALUES ARE NEGATIVE BECAUSE THE CONSTANT TERM HAS BEEN DROPPED.

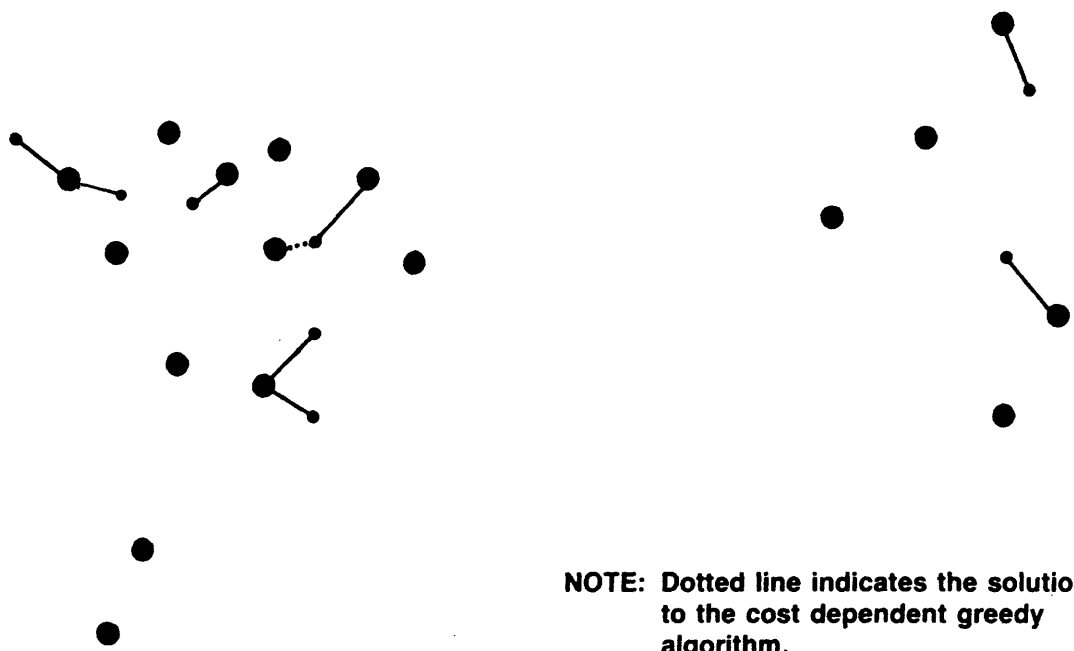
Figure 4. Branch and bound tree for N = 40, m = 20.



**Figure 5. Twenty five city example: educated eyeball solution with nine earth stations, \$17.8 million.**



**Figure 6. Twenty five city example: demand dependent greedy algorithm solution with nine earth stations, \$16.24 million.**



**NOTE: Dotted line indicates the solution to the cost dependent greedy algorithm.**

**Figure 7. Twenty five city example: optimal solution with 17 earth stations, \$14.96 million.**