FACULTY WORKING PAPER 91-0170

Solving Structured Multifacility Location Problems Efficiently

Dilip Chhajed
Department of Business Administration
University of Illinois

Timothy J. Lowe
Department of Management Science
University of Iowa

Bureau of Economic and Business Research
College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

BEBR

FACULTY WORKING PAPER NO. 91-0170

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

October 1991

Solving Structured Multifacility Location Problems Efficiently

Dilip Chhajed
Department of Business Administration
University of Illinois

Timothy J. Lowe
Department of Management
University of Iowa

Solving Structured Multifacility Location Problems Efficiently

Dilip Chhajed*

Timothy J. Lowe#

October 1991

Abstract

A generic multifacility location problem is considered which subsumes, as special cases, several NP-hard location problems that have appeared in literature. A unified algorithm is presented which solves the generic problem in polynomial time when the interactions between new facilities have special structure.

* Department of Business Administration, 350 Commerce West, University of Illinois at Urbana-Champaign, Champaign, IL 61820. email: Chhajed@uiucvmd.bitnet
# Department of Management Sciences, University of Iowa, Iowa City, IA 52242. email: Tlowe@opus-po.biz.uiowa.edu

1. Introduction

In this paper we consider a generic discrete multifacility location problem in which m new facilities are to be located. The locations of these new facilities are represented by decision variables $Y=\{y_1,\ldots,y_m\}$. Each y_p can take a value from the set $Z=\{z_1,\ldots,z_w\}$. Demands for services from the new facilities are located at points (called existing facilities) denoted by $v_1,v_2,\ldots,v_n$. The points $v_1,\ldots,v_n$ and $z_1,\ldots,z_w$ are in some metric space. The objective function of the location problem involves distances between pairs of facilities. Generally, the examples of the generic problem that we cite will be problems on networks (hence the use of v for existing facilities (at vertices)), but this need not be the case. For location problems on networks, the distance function, d(.), is taken to be the shortest path distance.

We first formulate the generic multifacility location problem (denoted by GMLP). Specific examples are given in section 2.

GMLP: $\quad$ OPT $F(y_1,\ldots,y_m)$

$\quad$ s.t. $y_i \in Z \qquad i=1,\ldots,m.$

In this problem, OPT is an optimization operator, either minimization or maximization. The function F takes the form $F(y_1,\ldots,y_m) = g(\{h_b(Y_b): b \in B\})$, where $Y_b \subseteq Y$, $\forall\ b \in B$ and g is either maximum, minimum or summation operator. For every $b \in B$, h_b is a function (maximum, minimum or summation) of one or more distance values. Each distance value is a weighted distance between some new facility and existing facility, e.g. $\alpha_{ip}d(v_i,y_p)$, or between a pair of existing facilities, e.g. $\beta_{pq}d(y_p,y_q)$.

Each of the examples of GMLP that we cite is an NP-hard optimization problem. Our focus in this paper is to specify a sufficient condition under which GMLP (and therefore the examples) is solvable in polynomial time. The condition for GMLP involves membership of the sets $Y_b, b \in B$ as follows. Construct an undirected *dependency graph* $G(V,E)$ with node set $V=\{y_1,\ldots,y_m\}$, and $(y_p,y_q) \in E$ if and only if both y_p and y_q are members of some Y_b, $b \in B$. We delete multiple edges from G. The dependency graph G has the following simple interpretation: If (y_p,y_q) is an edge of G, then an optimal value for the variable y_p (y_q) depends upon the value of y_q (y_p). If y_p is connected to several nodes in G, its optimal value depends upon all of its neighbors in G. If the dependency graph is a k-tree (to be defined later), then GMLP is solvable in polynomial time for a fixed value of k with an algorithm that exploits the structure of G.

As mentioned above, the domain of each function h_b in GMLP involves weighted distances between pairs of facilities. As we will see, in order to satisfy the tractability condition it will be necessary that some of the weights are zero. Thus for example, if $\alpha_{iq}=0$ for some i and q, then the weighted distance between existing facility i and new facility q has no affect on the location problem. Let $N=\{1,2,\ldots,n\}$ denote the index set of existing facilities (the v_i's) and $M=\{1,2,\ldots,m\}$ the index set of new facilities (y variables). In what follows for fixed $i \in N$ we denote by A_i the set of all q in M where $\alpha_{iq} \neq 0$. Similarly, for fixed $q \in M$, we denote by B_q the set of all p in M where $\beta_{pq} \neq 0$, $p \neq q$.

The remainder of the paper is organized as follows: In section 2 we give several discrete location problems, many of which have appeared in the literature. We then relate each example to GMLP by specifying OPT, g, B, as well as Y_b and h_b for every $b \in B$. In section 3 we discuss nonserial dynamic programming and indicate how it relates to GMLP. In section 3 we also define and discuss k-trees. Section 4 contains an algorithm for solving GMLP when G is a k-tree. For ease of explanation, we assume that k=3 but later on in section 4 we briefly outline how the algorithm changes when $k \neq 3$. Section 4 also contains our complexity analysis of the algorithm. In section 5, we provide an example indicating how the steps of the algorithm are implemented. Concluding remarks are given in section 6.

2. Examples of the Generic Multifacility Location Problem

We now give several examples of location problems which are instances of GMLP. Most of these problems have appeared in the literature. For the most part, we do not provide extensive motivation for the problems herein, since this motivation appears in the references that we cite. Some of the problems as *originally* stated in the literature will not fit our format. An example of this is the first problem below - the m-median problem. As originally stated, the m new facilities to be located are *indistinguishable* from each other in the sense that any new facility can provide service to a given existing facility, provided it is the closest. To fit our format, it is necessary that only a subset of the new facilities can provide service, and so the indices of the new facilities must be accounted for in the model formulation.

One of the keys to obtaining polynomial solvability of the versions of the models we study is that Z is a finite set. Recall that each y_p, $p \in M$ is restricted to be located at some point $z \in Z$. All of the problems that we cite are actually "continuous" problems in the sense that each new facility can be located anywhere on the network. Of course to fit our format, the solution set Z must be finite. Hooker, Garfinkel and Chen (1991) studied a large number of continuous network location problems in an effort to identify a finite set of points on the network which would contain the new facility locations in an optimal solution. They called such a set of points for a given problem a *finite domination set* (FDS). The first six problems that we cite have known FDS's of polynomial size and we indicate what this set is in each case. To the best of our knowledge, similar FDS results on problems 7-14 are not available except in very special cases. We emphasize that our focus is not on continuous network location problems and thus even if an FDS is known, we do not require that Z be a subset of the FDS.

In the problems to follow, relative to GMLP we specify OPT, g, B, Y_b and $h_b(Y_b)$. In problems 1-10, OPT is minimization and in problems 11-14, OPT is maximization.

Problem 1. m-Median Problem (Hakimi, 1964,1965)

$$\min \sum_{i=1}^{n} \min_{p \in A_i} \{\alpha_{ip} d(v_i, y_p)\} \tag{1}$$

In the context of GMLP, g is summation and B=N, so that for $b \in B$, $Y_b=\{y_q : q \in A_b\}$ and $h_b(Y_b) = \min_{p \in A_b} \{\alpha_{bp} d(v_b, y_p)\}$.

As originally stated in the literature, A_i=M for all i and $\alpha_{ip}=\alpha_i$ for all $p \in M$. It is necessary that A_i be a subset of M for our approach to apply. For this problem, Hakimi has shown that an FDS is the set of vertices of the network.

Problem 2. Two-Stage m-Median Problem (Goldman, 1971)

This generalization of the m-median problem involves movement of material between pairs of existing facilities. This movement is permitted to pass through one or two servers (new facilities) on the way. For a given ordered pair (i,j) with i and j in N, there are nonnegative weights α^1_{ij}, α^2_{ij} and α^3_{ij}, and a set $A_{ij} \subset M$.

$$\min \sum_{i=1}^{n} \sum_{j=1; j\neq i}^{n} \{\min_{p \in A_{ij},\, q \in A_{ij}} \{\alpha^1_{ij} d(v_i, y_p) + \alpha^2_{ij} d(y_p, y_q) + \alpha^3_{ij} d(y_q, v_j)\}\}. \tag{2}$$

For this problem, g=summation and B is the set of ordered pairs (i,j) where at least one of α^1_{ij}, α^2_{ij} or α^3_{ij} is non zero. For $b \in B$, $Y_b=\{y_q : q \in A_b\}$ and $h_{(i,j)}$ is the expression within the outer braces in (2). Goldman showed that an FDS for this problem is the set of vertices of the network.

Problem 3. Vector Assignment m-Median Problem (Weaver and Church, 1985)

In this variant of the m-median problem, an existing facility need not be served by a closest new facility. Instead, for existing facility i, f_{it} is the fixed fraction of service provided by the t^{th} closest new facility. With α_i as the total service required, define

$\alpha_{it}=\alpha_i f_{it}$. With A_i the set of indices of the subset of new facilities that can potentially provide service to existing facility i, we have

$$\min \sum_{i=1}^{n} \{ \sum_t \alpha_{it}\, t\text{-}\min_{p\in A_i}\{d(v_i,y_p)\}\}, \qquad (3)$$

where t-min is the t^{th} smallest value of $\{d(v_i,y_p):p\in A_i\}$.

Again, g=summation, B=N, $Y_b=\{y_q:q\in A_b\}$ and h_b is the expression within the outer braces in (3) above. Hooker, Garfinkel and Chen (1991) have shown that if $f_{it} \geq f_{i(t+1)}$ for all i and t, then an FDS is the set of vertices of the network.

Problem 4. Stochastic m-Median Problem (Mirchandani and Odoni, 1979)

In this problem, the length of each arc depends upon the "state of nature." There are a finite number of states, and in state r (which occurs with probability P^r) $d^r(x,y)$ is the shortest path distance between points x and y in the network. The problem then becomes

$$\min \sum_r P^r \sum_{i=1}^{n} \alpha_i \min_{p\in A_i}\{d^r(v_i,y_p)\}. \qquad (4')$$

Interchanging the order of summation in (4'), we have the equivalent problem

$$\sum_{i=1}^{n} \{\alpha_i \sum_r P^r \min_{p\in A_i}\{d^r(v_i,y_p)\}\}. \qquad (4)$$

Mirchandani and Odoni have shown that an FDS for (4) is the set of vertices of the network. The specifications of this problem are the same as problem 1 except that h_b is the expression within the outer braces in (4).

Problem 5. m-Median Problem with Mutual Communication (Dearing, Francis, and Lowe, 1976; Kolen, 1986; Fernandez-Baca, 1989; Chhajed and Lowe, 1990)

This problem is to minimize the sum of weighted distances between pairs of existing and new facilities as well as pairs of new facilities. As before, for i=1,...,n, $A_i\subset$

M. In addition, let ψ be the set of ordered pairs (p,q) where $\beta_{pq} \neq 0$. We assume that there are a total of τ pairs in ψ.

$$\min \sum_{i=1}^{n} \sum_{p \in A_i} \alpha_{ip} d(v_i, y_p) + \sum_{(p,q) \in \psi} \beta_{pq} d(y_p, y_q). \qquad (5')$$

Expression (5') can be written as

$$\min \sum_{p=1}^{m} \{ \sum_{i: p \in A_i} \alpha_{ip} d(v_i, y_p) \} + \sum_{(p,q) \in \psi} \beta_{pq} d(y_p, y_q). \qquad (5)$$

In (5), g is summation and letting $m+1, \ldots, m+\tau$ index the ordered pairs in ψ, $B=\{1,\ldots,m,m+1,\ldots,m+\tau\}$. For $b=1,\ldots,m$, h_b is the expression in braces in (5), with $Y_b=\{y_b\}$. For the remaining $b \in B$, $h_b = \beta_{pq} d(y_p, y_q)$ with $Y_b=\{y_p,y_q\}$ for $(p,q) \in \psi$. Kolen (1986) has shown that an FDS is the set of vertices of the network. Since the sets Y_b, for $b=1,\ldots,m$ generate no edges in the dependency graph G, we can take $A_i = M$ for all i. It is the membership of each Y_b, $b > m$, that is important in this problem.

Problem 6. m-Center Problem (Hakimi, 1965)

$$\min \{ \max_{i=1\ldots n} \{ \min_{p \in A_i} \{ \alpha_{ik} d(v_i, y_p) \} \} \} \qquad (6)$$

In this problem, B=N and $Y_b=\{y_q : q \in A_b\}$. h_b is the minimum of $|A_b|$ weighted distances, and g is the maximum of n values. As originally stated in the literature, $A_i=M$ for all i and $\alpha_{ip}=\alpha_i$, for all $p \in M$. However, we focus on version (6). For fixed $p \in A_i \cap A_j$, let x be a point (at a vertex in the interior of an arc of the network) which is on some path between vertices v_i and v_j and where $\alpha_{ip} d(v_i,x) = \alpha_{jp} d(v_j,x)$. Such a point is often referred to as a center bottleneck point (Hooker, Garfinkel and Chen, 1991). Let C be the union of all such points for all i,j and $p \in A_i \cap A_j$. It follows from the analysis of Handler and Mirchandani (1979), that an FDS is the set of vertices along with the points in C.

Problem 7. m-Center Problem with Mutual Communication (Dearing, Francis, and Lowe, *1976; Kolen, 1986; Chhajed and Lowe, 1990)*

This problem is similar to problem 5, except that instead of summing weighted distances, the maximum of all weighted distances is important.

$$\min \ \max\{\max_i\{ \max_{p\in A_i} \{ \alpha_{ip}d(v_i,y_p)\}\}; \max_{(p,q)\in\psi} \{ \beta_{pq}d(y_p,y_q)\}\}. \qquad (7')$$

Expression (7') can be rewritten as

$$\min \ \max \{\max_p\{ \max_{i:p\in A_i} \{\alpha_{ip}d(v_i,y_p)\}\}; \max_{(p,q)\in\psi} \{ \beta_{pq}d(y_p,y_q)\}\}. \qquad (7)$$

g is maximization, $B=\{1,\ldots,m+\tau\}$, where ψ has τ pairs. For $b=1,\ldots,m$, $Y_b=\{y_b\}$ with $h_b=\max_i\{\alpha_{ib}d(v_i,y_b):b\in A_i\}$. The remaining h_b's are $\beta_{pq}d(y_p,y_q)$ for $(p,q)\in\psi$ with $Y_b=\{y_p,y_q\}$. As in problem 5, it is the membership of the Y_b's, $b>m$, that is important in this problem. A_i can be all of M for every i.

Problem 8. Minimizing Operating Cost and Service Loss.

This problem has not been previously studied. Operating cost of the multifacility system is measured by the sum of weighted distances between pairs of facilities, as in problem 5. In addition, service loss for new facility p is proportional to the maximum distance to any customer that it serves. We allow possibly different weights, α^1_{ip} and α^2_{ip}, for the operating cost and service loss, respectively.

$$\min \sum_{p=1}^{m} \{ \sum_{i:p\in A_i} \alpha^1_{ip}d(v_i,y_p) + \max_{i:p\in A_i} \{\alpha^2_{ip}d(v_i,y_p)\}\} + \sum_{(p,q)\in\psi} \beta_{pq}d(y_p,y_q). \quad (8)$$

In this problem, B, g and Y_b are as in problem 5. h_b for $b=1,\ldots,m$ is the expression in the outer braces in (8). The remaining h_b's are the same as in problem 5.

Problem 9. m-Median Problem with Interacting Facilities

This is a new problem formed by allowing interaction between new facilities in the m-median problem. This can be written as:

$$\min \sum_{i=1}^{n} \min_{p\in A_i} \{\alpha_{ip} d(v_i,y_p)\} + \sum_{(p,q)\in\psi} \beta_{pq} d(y_p,y_q). \qquad (9)$$

In this problem g is summation and $B=\{1,\ldots,n+\tau\}$ where τ is the number of ordered pairs in ψ. For $b=1,\ldots,n$, $Y_b=\{y_q{:}q\in A_b\}$ and $h_b(Y_b) = \min_{p\in A_b} \{\alpha_{bp} d(v_b,y_p)\}$.The remaining τ h_b's are $\beta_{pq}(y_p,y_q)$ for $(p,q)\in\psi$ with $Y_b=\{y_p,y_q\}$.

Problem 10. m-Interacting Center

In this problem, which has not been studied before, the objective is to minimize the maximum (over all new facilities) of the total interaction of a new facility. The total interaction of a new facility is defined as sum of its interactions with existing and new facilities.

$$\min \max_p \{ \sum_{i:p\in A_i} \alpha_{ip} d(v_i,y_p) + \sum_{q\in B_p} \beta_{pq} d(y_p,y_q)\} \qquad (10)$$

In (10), g is maximization, B=M, $Y_b=y_p\cup\{y_q{:}q\in B_p\}$ for p=b and h_b is the expression in the braces in (10).

In the problems to follow, the new facilities to be located possess undesirable characteristics and so total system utility is nondecreasing in weighted distances between pairs of facilities. These problems are often called obnoxious or noxious facility location problems. A recent survey by Erkut and Neuman (1989) addresses many of these problems. In problems 11-14, OPT is maximization.

Problem 11. Maxisum Problem (Erkut, Baptie and Hohenbalken, 1990; Tamir, 1991)

This problem is the same as problem 5 except that the objective function is to be maximized.

$$\max \sum_{p=1}^{m} \{ \sum_{i:p\in A_i} \alpha_{ip} d(v_i,y_p)\} + \sum_{(p,q)\in\psi} \beta_{pq} d(y_p,y_q). \qquad (11)$$

See problem 5 for specifications of g, B, Y_b and h_b.

This problem with α_{ip}=0 for all i and p is called the defense-sum problem by Erkut and Neuman (1989), and has been studied by Kuby (1987), Hansen and Moon (1988), Erkut, Baptie and Hohenbalken (1990) and Erkut and Neuman (1990).

Problem 12. Anticenter Dispersion (Erkut, 1990; Tamir, 1991)

This problem is the same as problem 7 except max and min are interchanged in expression (7).

$$\max \min \{\min_p\{\min_{i:p\in A_i} \{\alpha_{ip} d(v_i,y_p)\}\}; \min_{(p,q)\in\psi} \{\beta_{pq}(y_p,y_q)\}\}. \tag{12}$$

The sets B and Y_b are as in problem 7, but g is minimization. For b=1,...,m, $h_b=\min_i\{\alpha_{ib}d(v_i,y_b):p\in A_i\}$ with $Y_b=\{y_b\}$. The remaining h_b's are as in problem 7. Also, A_i can be all M for every i.

In the above problem, when $\alpha_{ip}=\infty$ for all i and p, the resulting problem is called the dispersion problem. The dispersion problem has been studied by Shier (1977), Chandrasekaran and Daughety (1981), Tansel, Francis, Lowe and Chen (1982), Chandrasekaran and Tamir (1982), Kuby (1987), Erkut (1990) and Erkut and Neuman (1989, 1990).

Problem 13. Dispersion Sum (Erkut and Neuman, 1990).

$$\max \min_p\{\sum_{i:p\in A_i} \alpha_{ip} d(v_i,y_p) + \sum_{q\in B_p}\beta_{pq} d(y_p,y_q)\}. \tag{13}$$

Erkut and Neuman study a version of this problem where $\alpha_{ip}=\infty$ for all i and p. We give the specifications for the problem as stated in (13). In (13), g is minimization, B=M, $Y_b=y_p\cup\{y_q:q\in B_p\}$ for b=p and h_b is the sum in the braces in (13).

Problem 14. Defense (Erkut and Neuman, 1990).

$$\max \sum_{p=1}^{m} \min\{\{\alpha_{ip}d(v_i,y_p):p\in A_i\}\ ;\ \{\beta_{pq}d(y_p,y_q):\ q\in B_p\}\}. \quad (14)$$

As in problem 13, Erkut and Neuman study the case with $\alpha_{ip}=\infty$ for all i and p. In (14), we note that g is summation, B=M, $Y_b=y_p\cup\{y_q:q\in B_p\}$ for p=b and h_b is the minimum of the terms in the braces in (14).

3. Variable Elimination

3.1 GMLP and Non-Serial Dynamic Programming

Consider the following problem,

$$\text{(P)} \qquad \min F(Y) = \min \textstyle\sum_{b\in B} h_b(Y_b),$$

where $Y = \{y_1,\ldots y_n\}$ is a set of discrete variables, B is a finite index set, and $Y_b \subseteq Y$. Problem (P) is an instance of (GMLP) with OPT = minimization and g the summation operator.

When $Y_b = \{y_i,y_{i+1}\}$ for some i=1,...,n-1, (P) is a serial unconstrained problem which can be solved by the usual dynamic programming method. When $Y_b\subset Y$, (P) is a non-serial dynamic programming problem (Bertele and Brioschi, 1972) and can be solved efficiently by successive variable elimination in certain cases as we will describe shortly.

Returning to (GMLP), recall that g is either maximum, minimum or summation, so that g is both separable and monotone nondecreasing in each of its arguments. Thus g is *decomposable* (Minoux, 1986, Section 9.2), which is sufficient to justify the following approach to solving (GMLP).

The idea of variable elimination is to replace the original problem with a new problem involving fewer variables, but where the new problem is the same as the original. Suppose $Y^e \subset Y$ is to be eliminated. In (GMLP), let

$B(Y^e) = \{b\in B:h_b$ is a nonconstant function of at least one member of $Y^e\}$.

Define $\bar{Y}^e = \{\{\cup_{b\in B(Y^e)} Y_b\}\backslash Y^e\}$,

$F_1(Y^e,\bar{Y}^e) = g(\{h_b(Y_b):b\in B(Y^e)\})$, and

$F_2(\bar{Y}^e) = OPT_{y\in Y^e}(F_1(Y^e,\bar{Y}^e))$.

We note that $F_1(Y^e,\bar{Y}^e)$ is the result of applying g to a subset , namely all $b\in B(Y^e)$, of functions $\{h_b\}$, $F_2(\bar{Y}^e)$ is the result of optimizing $F_1(.,.)$ over $y \in Y^e$ with $\bar{Y}^e$ fixed. Thus for any choice of g and OPT, we have, with

$$F'(Y\backslash Y^e) \equiv g(F_2(\bar{Y}^e), \{h_b(Y_b):b\notin B(Y^e)\}),$$

$$OPT\ F(y_1,\ldots y_m) = OPT\ \ F'(Y\backslash Y^e).$$

For this approach to work efficiently we need,

(i) $\bar{Y}^e$ to be a proper subset of $Y\backslash Y^e$. Otherwise the procedure amounts to complete enumeration.

(ii) A partition of variables in sets and ordering of these sets reflecting the order in which variables are to be eliminated. This latter problem is called the *secondary optimization problem* in non-serial dynamic programming (Bertele and Brioschi, 1972).

It is easy to see that if the dependency graph G is a complete graph, then $\bar{Y}^e = Y\backslash Y^e$ for any choice of Y^e and the variable elimination approach will not be effective for this case. This would occur for example, if for every pair (y_p,y_q) there is some $b\in B$ where $\{y_p,y_q\} \subset Y_b$.

Given a new problem, it is possible that a reformulation may be required to make the problem amenable to our method or to make the problem easier to solve. For example, problem 5 can also be stated as,

$$\min \sum_{k=1}^{m} \{ \sum_{i:k\in A_i} \alpha_{ik}d(v_i,y_k) + \sum_{l\in B_k;l>k}\beta_{kl}d(y_k,y_l)\}, \qquad (5'')$$

with $B=\{1,\ldots,m\}$, g is summation, h_b is the term in braces in (5") and for b=k, $Y_b=B_k$. But this formulation results in a dependency graph which is a supergraph of the dependency graph of (5) and may require more effort if solved by our method.

We next define a class of graphs for which if the dependency graph is in this class, it will guarantee that an efficient variable elimination approach will work for GMLP.

3.2 K-Trees

A *k-clique* is complete graph on k-vertices. A *k-tree* is recursively defined as follows: A k-clique is a k-tree. Given a k-tree and a subgraph of the k-tree which is a k-clique, the graph obtained by introducing a new node and connecting it to every node of the k-clique is again a k-tree. Subgraphs of k-trees are also referred to as *partial k-trees*.

A node y with degree k is a *k-leaf* if all of the k nodes adjacent to it (neighbors) induces a k-clique. A k-leaf along with its neighbors forms a (k+1)-clique. If we eliminate a k-leaf of a k-tree, the resulting graph is again a k-tree. Thus, repeated elimination of k-leaves will result finally in a graph which is a k-clique. For a k-tree with $m > k$ vertices, let the set $Y_S = \{y_1,\ldots,y_{m-k}\}$ denote the ordering of vertices in an elimination sequence of k-leaves and let the k remaining nodes of the k-tree be arbitrarily numbered $y_{m-k+1},\ldots,y_m$. Thus y_1 represents the node which is eliminated first, followed by y_2, y_3, and so on, until y_{m-k} is eliminated after which we obtain a k-clique formed by nodes $y_{m-k+1},\ldots,y_m$. Note that y_{t+1}, $t < m-k$, may not be a k-leaf in the original k-tree but will be a k-leaf after y_t is eliminated.

Suppose that graph G is a k-tree, and let G_t denote the graph immediately before node y_t is eliminated. Also let Q_t be the k-clique adjacent to node y_t in the graph G_t. Given a reduction sequence Y_S, the stage t *descendents* D_t is the set composed of y_t along with all nodes $y_j \in Y$, $j < t$, such that when y_j was removed in the elimination sequence, each node to which it was adjacent at the time of its removal, i.e. adjacent in G_j, was either a member of Q_t or a member of D_t. We note that unlike Arnborg and Proskurowski (1989), our definition of descendents does not include any nodes y_k with $k > t$. Note that Q_t may be equal to Q_j, for some $j \neq t$ (i.e., y_t and y_j are adjacent to the same k-clique when eliminated) but D_t will not be equal to D_j in such cases, since if $j<t$, then y_t is by definition not in D_j. In the graph G_t, node y_t is the only node of D_t which is in G_t, as all other nodes of D_t have

been eliminated when G_t is obtained.

k-trees form a rich class of graphs and contain several well known graph families. A tree, defined in the usual sense, is a 1-tree. Series-parallel graphs, circuits, outer-planar graphs, and cactus graphs are all partial 2-trees. A Halin graph is a partial 3-tree. Many combinatorial problems have been solved when restricted to partial k-trees (Corneil and Keil, 1987; Takamizawa, et al., 1982; Arnborg and Proskurowski, 1989; Fernandez-Baca, 1989).

Although efficient algorithms exist for several problems when restricted to partial k-trees, recognizing whether a graph is a partial k-tree remains a difficult problem. Duffin (1965) gave a characterization for partial 2-trees as graphs with no subgraphs homeomorphic to K_4. An efficient algorithm for recognizing partial 2-trees is given in Wald and Colbourn (1983). Such a characterization of k-trees is not found for $k \geq 3$. However, a partial 3-tree can be recognized and embedded on a 3-tree in $O(m^3)$ time by the algorithm given by Arnborg and Proskurowski (1986), which they indicate can be improved to O(mlogm). For a fixed value of k, recognizing partial k-trees and embedding them in a k-tree, if such an embedding exists, can be done in $O(m^{k+2})$ (Arnborg and Proskurowski, 1987). Lagergren (1990) has given, for a fixed k, an $O(\log^3 m)$ parallel algorithm for finding the tree-decomposition, which can be implemented in $O(m\log^2 m)$ time in a sequential manner.

4. GMLP on K-Trees

In this section we first present, in detail, an algorithm to solve GMLP when the dependency graph G is a 3-tree. Subsequently, we discuss the k-tree case, where $k>3$.

4.1 GMLP on 3-Trees

We have chosen to concentrate on 3-trees because they are easy to recognize, and k being small will keep our exposition simple. Our assumption that the dependency graph is a 3-tree (as opposed to partial 3-tree) is of no consequence since artificial edges (Arnborg and Proskurowski, 1987) can be added to a partial 3-tree to complete it to a 3-tree. We note that when the dependency graph is a 3-tree, every function h_b is a function of no more than four y variables. This follows since a k-tree has no clique of size larger than k+1.

Note that the *range* of each function $h_b(Y_b)$ can be represented as a table of values in which for a particular instantiation of the variables in Y_b we record the value of the function. For example if $Y_b=\{y_u,y_p,y_q\}$, then $h_b(y^\circ_u,y^\circ_p,y^\circ_q)$ represents the value of h_b when $y_u = y^\circ_u$, $y_p=y^\circ_p$, and $y_q=y^\circ_q$. In what follows we suppress the b index of the functions $h_b(.)$ when no confusion results.

During the algorithm as variables are eliminated, we compute new functions (denoted by h^r) which represent the value of an *optimal* solution to a modified GMLP, restricted to a subset of the variables Y. In essence, the functions h^r are like the recursion functions in ordinary dynamic programming. We refer to all such functions as *r-functions.*

Suppose at some stage t, variable y_t is to be removed. With Q_t the 3-clique adjacent to y_t in G_t, $h^r(Q^\circ_t)$ represents the *value of an optimal solution* to GMLP, restricted to variables in D_t, given that each $y_k \in Q_t$ is *fixed* at y°_k. The computation of $h^r(Q^\circ_t)$ involves a) each original h function whose domain includes y_t and is a subset of $y_t \cup Q_t$, and b) each previously computed r-function which has as a domain a three variable subset of $y_t \cup Q_t$, and where each variable is of course, fixed at a specific value. At the beginning of the algorithm, every r-function is initially set to INT. "Updates" of r-functions occur during the course of the algorithm.

In addition to updating r-functions at each iteration, we also update label sets. The label set $L(Q^\circ_t)$, denotes the *values* of the *variables* in D_t which attain the objective $h^r(Q^\circ_t)$. These label sets are used to "trace out" an optimal solution to GMLP once the algorithm terminates. At the beginning of the algorithm, all label sets are initialized to be empty.

We now give the (update) formulas for $h^r(.)$ and $L(.)$. At stage t (when y_t is eliminated) suppose that $Q_t = \{y_u, y_p, y_q\}$. Let y°_u, y°_p, y°_q be fixed values of the variables y_u, y_p, and y_q respectively. Then ,

$$h^r(y^\circ_u,y^\circ_p,y^\circ_q) = g(h^r(y^\circ_u,y^\circ_p,y^\circ_q), \mathrm{OPT}_{y_t \in Z} (g(h(y_t,y^\circ_u,y^\circ_p,y^\circ_q), h(y_t,y^\circ_p,y^\circ_q), h(y^\circ_u,y_t,y^\circ_q), h(y^\circ_u,y_t,y^\circ_p), h(y^\circ_u,y_t), h(y_t,y^\circ_p), h(y_t,y^\circ_q), h(y_t), h^r(y_t,y^\circ_p,y^\circ_q), h^r(y^\circ_u,y_t,y^\circ_q), h^r(y^\circ_u,y_t,y^\circ_p))), \quad (15)$$

and

$$L(y^\circ_u,y^\circ_p,y^\circ_q)=y^*_t \cup L(y^*_t,y^\circ_p,y^\circ_q) \cup L(y^\circ_u,y^*_t,y^\circ_q) \cup L(y^\circ_u,y^*_t,y^\circ_p) \cup L(y^\circ_u,y^\circ_p,y^\circ_q), \quad (16)$$

where y^*_t attains the OPT in (15). The domain of the inner optimization in (15) includes all h and h^r functions whose domains contain y_t and are contained in $y_t \cup Q_t$. Note that all variables are fixed except variable y_t, and if for some subset $Y' \subset y_t \cup Q_t$ there are two or more h functions with domain Y', all such functions are included in the optimization. The update formula (15) is written with the understanding that if a given problem GMLP does not have all of the h functions listed in the inner minimization, such functions should be deleted from (15). Also, we use the convention that any $h^r(.)$ function on the right hand side of (15) whose current value is INT (its initialized value) should be deleted from the right hand side of (15). (For any *specific instance* of GMLP (choice of OPT and g) an appropriate numerical value of INT can be used, e.g., zero, a very large number, or a very small number. INT is chosen so that its *value* does not affect any update. Since the algorithm is for a general GMLP, we employ the "deletion" convention.) Finally, we note that in computing $h^r(y^\circ_u,y^\circ_p,y^\circ_q)$, we have not used $\{h(y^\circ_u,y^\circ_p,y^\circ_q), h(y^\circ_u,y^\circ_p), h(y^\circ_u,y^\circ_q), h(y^\circ_p,y^\circ_q), h(y^\circ_u), h(y^\circ_u), h(y^\circ_u)\}$. These functions will be accounted for when y_u, y_p and y_q are removed at a later stage.

We now give the algorithm followed by a proof of its correctness.

Algorithm 3TREE

Step 1: Find an elimination ordering Y_S of the nodes of G and renumber the nodes of G according to the sequence in Y_S.

Set $t:=1$, $G_t=G$ and all $h^r(.) = \text{INT}$, $L(.) = \varnothing$.

Step 2: Let $Q_t = \{y_u, y_p, y_q\}$.

For all y°_u, y°_p and y°_q,

Set $h^r(y^\circ_u, y^\circ_p, y^\circ_q)$ and $L(y^\circ_u, y^\circ_p, y^\circ_q)$ using (15) and (16), respectively.

Set $G_{t+1} = G_t \backslash y_t$. Reset $t=t+1$.

Step 3: If $t \leq m-3$, repeat Step 2. Else go to Step 4.

Step 4: Let (y_u, y_p, y_q) be the three nodes remaining in G_{m-2}.

Find $C(y^*_u, y^*_p, y^*_q) = \text{OPT}(g(h^r(y^\circ_u, y^\circ_p, y^\circ_q), h(y^\circ_u, y^\circ_p, y^\circ_q), h(y^\circ_u, y^\circ_p), h(y^\circ_p, y^\circ_q), h(y^\circ_u, y^\circ_q), h(y^\circ_u), h(y^\circ_q), h(y^\circ_p))$ for all y°_u, y°_p and $y^\circ_q \in Z)$.

$L(y^*_u, y^*_p, y^*_q)$ along with y^*_u, y^*_p and y^*_q will give the values of the y variables that obtains the OPT solution value $C(y^*_u, y^*_p, y^*_q)$.

To show the correctness of the algorithm we need the following Lemma, which is easily proven using the definition of descendents and the properties of 3-trees. Consider some arbitrary stage $t > 1$ of the algorithm, with $Q_t = \{y_u, y_p, y_q\}$. If $\{y_u, y_p, y_q\}$ was a 3-clique adjacent to a 3-leaf in the elimination ordering at some previous stage, let $T_{upq} < t$ be the largest stage index number when this was the case. Similarly if $\{y_t, y_p, y_q\}$, $\{y_t, y_u, y_p\}$, and $\{y_t, y_u, y_q\}$ were 3-cliques adjacent to 3-leaves in the elimination ordering at some previous iteration(s), define T_{tpq}, T_{tup}, $T_{tuq} < t$ as the largest stage indices when this was the case. In any of these cases, if a 3-clique $\{y_\alpha, y_\beta, y_\gamma\}$ was not adjacent to a 3-leaf, take $D_{T_{\alpha\beta\gamma}} = \varnothing$ in the statement of the Lemma.

<u>Lemma 1:</u> At stage $t>1$ with $Q_t = \{y_u, y_p, y_q\}$, we have

a) Let D_α and D_β be distinct nonempty sets from $D_{T_{upq}}$, $D_{T_{tpq}}$, $D_{T_{tup}}$ and $D_{T_{tuq}}$. Then

(i) $D_\alpha \cap D_\beta = \varnothing$,

(ii) with $y_i \in D_\alpha$ and $y_j \in D_\beta$, y_i and y_j are not adjacent in G.

b) $D_t = \{y_t\} \cup \{D_{T_{tpq}} \cup D_{T_{tup}} \cup D_{T_{tuq}} \cup D_{T_{upq}}\}$.

We are now ready to prove the correctness of step 2. In the following Theorem, we need to identify specific stages when *updates* of r-functions and label sets occur. Thus when we write $h^r{}_\tau(Q^\circ{}_\tau)$ and $L_\tau(Q^\circ{}_\tau)$ we mean the values of the r-functions and the label sets computed at stage τ.

<u>Theorem 1:</u> For $t \leq m-3$, $h^r{}_t(y^\circ{}_u,y^\circ{}_p,y^\circ{}_q)$ $(L_t(y^\circ{}_u,y^\circ{}_p,y^\circ{}_q))$ as determined in step 2 of 3TREE is the optimal value (solution) of (GMLP) restricted to new facilities in D_t with the locations $y^\circ{}_u$, $y^\circ{}_p$, and $y^\circ{}_q$ are fixed.

<u>Proof:</u> We prove this by induction on t. When t=1, $D_1 = \{y_t\}$. Since initially all $h^r(.)$ have value INT, no $h^r(.)$ functions appear on the right hand side of (15) and so,

$$h^r{}_1(y^\circ{}_u,y^\circ{}_p,y^\circ{}_q) = \mathrm{OPT}_{y_t \in Z}(g(h(y_t,y^\circ{}_u,y^\circ{}_p,y^\circ{}_q), h(y_t,y^\circ{}_p,y^\circ{}_q), h(y^\circ{}_u,y_t,y^\circ{}_q),$$
$$h(y^\circ{}_u,y_t,y^\circ{}_p), h(y^\circ{}_u,y_t), h(y_t,y^\circ{}_p), h(y_t,y^\circ{}_q), h(y_t)),$$

which is the value of the optimal solution over D_1 with $y^\circ{}_u$, $y^\circ{}_p$, and $y^\circ{}_q$ fixed and does not include functions with domain contained in $\{y^\circ{}_u,y^\circ{}_p,y^\circ{}_q\}$. Also since all label sets are initially empty, $L_1(y^\circ{}_u,y^\circ{}_p,y^\circ{}_q) = \{y^*{}_t\}$, where $y^*{}_t$ attains OPT.

We now assume that the theorem is true for up to t-1 iterations of step 2. At iteration t with $Q_t = \{y_u,y_p,y_q\}$, if $\{y_\alpha,y_\beta,y_\gamma\}$ (a subset of $\{y_t,y_u,y_p,y_q\}$) was not a 3-leaf in the elimination process at a prior iteration, then $h^r(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma) = \mathrm{INT}$ and $L(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma) = \varnothing$ for all fixed values of y_α, y_β and y_γ. This follows since no updates from the initialized values has occurred. Otherwise, with $T_{\alpha\beta\gamma}$ as defined in the discussion prior to Lemma 1, from the algorithm we have that the $h^r(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma)$ and $L(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma)$ were last updated in stage $T_{\alpha\beta\gamma}$ and have current values = $h^r_{T_{\alpha\beta\gamma}}(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma)$ and $L_{T_{\alpha\beta\gamma}}(y^\circ{}_\alpha,y^\circ{}_\beta,y^\circ{}_\gamma)$, respectively. Lemma 1 now justifies the computation of

$h^r(y^\circ_u, y^\circ_p, y^\circ_q)$ and the label $L(y^\circ_u, y^\circ_p, y^\circ_q)$ at stage t in step 2 of the algorithm. «»

Corollary: Algorithm 3TREE solves (GMLP) when the dependency graph is a 3-tree.

Proof: From Theorem 1, $h^r(y^\circ_u, y^\circ_p, y^\circ_q)$ is the optimal value of the (GMLP) restricted to D_{m-3} with fixed values y°_u, y°_p and y°_q at the last iteration of Step 2 when y_{m-3} is eliminated from the graph. Step 4 accounts for the remaining three variables, which gives the solution to (GMLP). «»

To derive the running time complexity of the algorithm 3TREE, we first assume that the functions h_b are available in the form of tables. We also assume that evaluating $g(\omega)$ takes $O(|\omega|)$ time, where ω is a row vector with cardinality $|\omega|$. This assumption is certainly valid for the forms of g that we consider. Clearly OPT(ω) takes $|\omega|$ time as well.

Using the algorithm of Rose, Tarjan and Lueker (1976), finding a reduction sequence of a 3-tree on m nodes takes O(3m)=O(m) time - the complexity of step 1 of 3TREE. For a fixed value of t, a total of w^3 r-functions are updated via (15). Each such update involves an inner optimization (over y_t), followed by an application of g to two values. Referring to the inner optimization, for fixed y_t, g is applied to a total of a) no more than three r-functions (those r-functions with values unequal to INT) and, b) a *subset* of original h functions (let λ_t denote the cardinality of this latter subset of functions). Since there are a total of w distinct values of y_t, the inner optimization takes $O(w(3+\lambda_t))$. The outer application of g, to two values, takes constant time. Since λ_t remains constant for fixed t, it now follows that step 2, for fixed t, takes $O(w^4(3+\lambda_t))$. We now note that each original h function is used in at most one stage t (it may appear in step 4). Thus summing the λ_t over stages 1 to m-3 we have $\sum_{t=1}^{m-3} \lambda_t \leq |B|$. It now follows that the total effort for step 2, over all stages, is $O(w^4(3m+|B|))$.

Finally, since step 4 can be done in $O(w^3)$ time, we have a total complexity of $O(w^4(3m+|B|))$ for 3TREE.

As a final note in this section, we observe that constrained versions of GMLP can be handled with our algorithm. For example, Francis, Lowe, and Ratliff (1978) consider a version of problem 1 where there are upper bounds on distances between pairs of facilities ((new, existing) as well as (new,new)). These constraints can easily be incorporated in the functions $h_b()$. For example, for problem 5 if locating the new facility u at z_i and new facility q at node z_j violates an upper bound involving new facilities u and q, then set the corresponding $h(y_u,y_q)$ (with $y_u=z_i$ and $y_q=z_j$) to be prohibitively large.

4.2 GMLP on K-Trees

The modification of the algorithm for k-trees, k>3 is straightforward. Step 1 remains the same. In a k-tree, the clique Q_t adjacent to y_t at the time of its removal has a total of k nodes. To perform step 2, the g function in the inner optimization of (15) is applied at a) all r-functions, with value unequal to INT, and whose domain includes y_t along with k-1 of the variables in Q_t; and b) a total of λ_t h functions, each of whose domains are in $y_t \cup Q_t$ and include y_t. We note that the number of r-functions in a) is no more than k. Step 2 is repeated m-k times, i.e. the terminal clique for step 4 has cardinality k.

Using the algorithm of Rose, Tarjan, and Lueker (1976), step 1 can be performed in $O(m+|E|)=O(km)$ time. The dominating effort for the k-tree case is in step 2. For fixed t, the effort for all $h^r(.)$ updates takes $O(w^{k+1}(k+\lambda_t))$. Since, as in the 3-tree case, $\sum_{t=1}^{m-k}\lambda_t \le |B|$, it follows that the total effort for step 2 is $O(w^{k+1}(mk+|B|))$, which is the effort for the algorithm.

5. An Example

We now give an example to illustrate the steps in the algorithm. Consider the m-median problem (problem 1) with 5 new facilities to be located and 5 existing facilities

,$\{v_1,v_2,\ldots,v_5\}$. There are two candidate location points, so $Z=\{z_1,z_2\}$. Facility 1 interacts with new facilities 2, 3 and 5, thus $A_1=\{y_2,y_3,y_5\}$. Other interactions are given by $A_2=\{y_2,y_5\}$, $A_3=\{y_1,y_3,y_4,y_5\}$, $A_4=\{y_1,y_3\}$ and $A_5=\{y_2,y_3,y_4\}$. The values α_{ip} and $d(v_i,z_j)$ are given in Figure 1. We will have five h functions with $Y_1=\{y_2,y_3,y_5\}$, $Y_2=\{y_2,y_5\}$, $Y_3=\{y_1,y_3,y_4,y_5\}$, $Y_4=\{y_1,y_3\}$ and $Y_5=\{y_2,y_3,y_4\}$. The dependency graph is shown in Figure 2 while the h functions, represented as tables, are given in Figure 3. Note that the dependency graph in Figure 2 is a 3-tree. We use the reduction sequence $\{y_1,y_3\}$. We now give details of step 2 and the termination step (step 4).

Step 2: Iteration 1:

The variable eliminated at this step is y_1. Thus, $y_t=y_1$ and $Q_t = \{y_3,y_4,y_5\}$. Relevant h functions for the right hand side of (15) are $\{h_3(.),h_4(.)\}$. The function $h^r(y_3,y_4,y_5)$, computed at this step is given in Figure 4. Eliminate y_1 and repeat step 2.

Step 2: Iteration 2:

y_3 is eliminated at this step, so $y_t=y_3$, $Q_t = \{y_2,y_4,y_5\}$. Relevant h functions are $\{h_1(.),h_5(.),h^r(y_3,y_4,y_5)\}$. The output of this step is $h^r(y_2,y_4,y_5)$ which is given in Figure 5. Eliminate y_3, and since only three nodes remain, go to step 4.

Step 4: Termination Step

$\{y_2,y_4,y_5\}$ are the remaining variables. Relevant h functions to be used in the g operator are $\{h^r(y_2,y_4,y_5), h_2(.)\}$. Figure 6 gives the value of the objective function for various choices for variables y_2,y_4 and y_5. From this figure we can identify two optimal solutions: $\{y_1=z_1,y_2=z_2,y_3=z_2,y_4=z_1,y_5=z_1 \text{ or } z_2\}$ with objective function value 21.

6. Conclusion

In this paper we have introduced a generic multifacility location problem (GMLP) which subsumes several well known location problems. In addition, three new multifacility location problems are introduced. By appropriately defining the dependency graph (perhaps after reformulation of the problem) and exploiting special structure of this graph, we obtained a polynomial time algorithm for GMLP when the dependency graph is a k-tree. The algorithm can be used to solve each one of the example problems, when k-tree structure is present, by applying the appropriate g function and optimization operator. This work should be useful to researchers in the field of location theory in the following way.

Given a new multifacility location problem, first determine whether it has an FDS. If the problem is naturally a discrete location problem (e.g., locations restricted to nodes) this step is unnecessary. The objective here is to find the set Z. We note that it is unnecessary that every variable y take on a value from the same finite set Z. In fact one can define a finite set Z(y) for every variable y. The next task is to determine the functions $h_b(Y_b)$. Since the edges of the dependency graph G are determined by variable pairs in the sets Y_b, the key is to find a valid formulation where G is as sparse as possible. (See our discussion on this point regarding problem 5 in section 3.1.) Then a check is made to determine whether G is a k-tree. For *fixed* k, a polynomial time algorithm is available to do this (Arnborg and Proskurowski, 1987). Finally, if a k-tree is found and g is decomposable, the algorithm given in this paper leads to a polynomial time algorithm for problem solution.

Acknowledgement

We would like to thank Charles Blair and Arie Tamir for their helpful comments.

References

Arnborg, S. and A. Proskurowski, “Linear Time Algorithms for NP-Hard Problems Restricted to Partial K-Trees,” Discrete Applied Mathematics, 23, (1989), 11-24.

Arnborg, S. and A. Proskurowski, “Complexity of Finding Embeddings in a K-Tree,” SIAM. J. Alg. Disc. Meth., 8, (1987), 277-284.

Arnborg, S. and A. Proskurowski, “Characterization and Recognition of Partial 3-Trees,” SIAM. J. Alg. Disc. Meth., 7, (1986),305-314.

Bertele, U. and F. Brioschi, "Nonserial Dynamic Programming," (1972), Academic Press.

Chandrasekaran, R. and A. Daughety, "Location on Tree Networks: P-Center and P-Dispersion Problems," Math. Oper. Res., 6, (1981), 50-57.

Chandrasekaran, R. and A. Tamir, "Polynomially Bounded algorithms for Locating p-Centers on a Tree, " Math. Prog., 22, (1982), 304-315.

Chhajed, D. and T. J. Lowe, “M-median and M-center Problems with Mutual Communication: Solvable Special Cases,” BEBR Paper No. 90-1654, University of Illinois at Urbana-Champaign, Champaign, IL, (1990), to appear in Operations Research.

Corneil, D.G. and J.M. Keil, “A Dynamic Programming Approach to the Dominating Set Problem on k-Trees,” SIAM. J. Alg. Disc. Meth., 8, (1987),535-543.

Dearing, P. M, R. L. Francis, and T. J. Lowe, “Convex Location Problems on Tree Networks,” Operations Research, 24, (1976), 628-642.

Duffin, R. J., "Topology of Series-Parallel Networks," J. Math. Anal. Appl., 10, (1965), 303-318.

Erkut, E.," The Discrete p-Dispersion Problem," (1990), to appear in Eur. J. Opnl. Res.

Erkut, E., T. Baptie and B.v. Hohenbalken, "The Discrete p-Maxian Location Problem," Comp. and Opnl. Res., 17, (1990), 51-61.

Erkut, E. and S. Neuman, "Analytical Models for Locating Undesirable Facilities," European Journal of Operations Research, 40, (1989), 275-291.

Erkut, E. and S. Neuman, "Comparison of Four Models for Dispersing Facilities," (1990), to appear in INFOR.

Fernandez-Baca, D. ,"Allocating Modules to Processors in a Distributed System," IEEE Trans. Software Eng., 15, (1989), 1427-1436.

Francis, R. L., T. J. Lowe, and H. D. Ratliff, "Distance Constraints for Tree Network Multifacility Location Problems," Operations Research, Vol. 26, 1978, pp. 570-596.

Goldman, A.J.," Optimum Location for Centers in a Network," Transportation Science, 5, (1971), 212-221.

Hakimi, S.L.,"Optimal Location of Switching Centers and the Absolute Centers and the Medians of a Graph," Operations Research, 12, (1964), 450-459.

Hakimi, S.L., "Optimum Distribution of Switching Centers in a Communications Network and Some Related Graph-Theoretic Problems," Operations Research, 13, (1965), 462-475.

Handler, G.Y., and P.B. Mirchandani, "Location on Networks: Theory and algorithms," MIT Press, Cambridge, Mass., (1979).

Hansen, P. and I.D. Moon, "Dispersing Facilities on a Network, " RUTCOR Research Report # 52-88, Rutgers University, New Brunswick, NJ, (1988).

Hooker, J.N., R.S. Garfinkel, and C.K. Chen, "Finite Dominating Sets for Network Location Problems," Operations Research, 39, (1991), 100-118.

Kolen, A.W.J., "Tree Network and Planar Rectilinear Location Theory," CWI Tract 25, Center for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands, (1986).

Kuby, M.J., "Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems," Geogr. Analy., 19, (1987), 315-329.

Lagergren, J., "Efficient Parallel Algorithms for Tree-Decomposition and Related Problems," Proc. of IEEE Conference on Foundations of Computer Science, (1990), 173-182.

Minoux, M., "Mathematical Programming," (1986), John Wiley and Sons.

Mirchandani, P.B. and A.R. Odoni, "Locations of Medians on Stochastic Networks," Transportation Science, 13, (1979), 85-97.

Rose, D.J., R. E. Tarjan, and G.S. Lueker, "Algorithmic Aspects of Vertex Elimination on Graphs," SIAM J. Comput., 5, (1976), 266-283.

Shier, D.R.,"A Min-Max Theorem for p-Center Problem on a Tree," Transportation Science,11, (1977), 243-252.

Takamizawa, K., T. Nishizeki, an S. Saito, "Linear-Time Computability of Combinatorial Problems on Series-Parallel Graphs," J. of the ACM, 29, (1982), 623-641.

Tamir, A., "Obnoxious Facility Location on Graphs," SIAM J. Disc. Math., to appear, (1991).

Wald, J. A. and C. J. Colbourn, "Steiner Trees, Partial 2-Trees, and Minimum IFI Networks," Networks, (1983), 159-167.

Weaver, J.R., and R.L. Church, "A Median Location Model with Nonclosest Facility Service," Transportation Science, 19, (1985), 58-74.

$d(v_i,z_j)$	z_1	z_2
v_1	2	1
v_2	1	5
v_3	1	3
v_4	2	1
v_5	3	4

α_{ip}	y_1	y_2	y_3	y_4	y_5
v_1		4	5		4
v_2		6		2	
v_3	2		2	4	3
v_4	8		7		
v_5		3	2	2	

Figure 1. Data for the Example

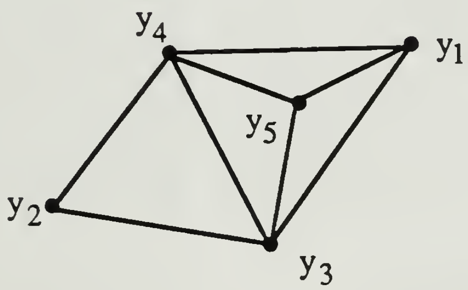

Figure 2. Dependency Graph for the Example Problem

y_2	y_3	y_5	$h_1(.)$
1	1	1	8
2	1	1	4
1	2	1	5
2	2	1	4
1	1	2	4
2	1	2	4
1	2	2	4
2	2	2	4

a) $h_1(y_2,y_3,y_5)$

y_2	y_3	y_4	$h_5(.)$
1	1	1	6
2	1	1	6
1	2	1	6
2	2	1	6
1	1	2	6
2	1	2	6
1	2	2	8
2	2	2	8

b) $h_5(y_2,y_3,y_4)$

y_2	y_4	$h_2(.)$
1	1	2
2	1	2
1	2	6
2	2	10

c) $h_2(y_2,y_4)$

y_1	y_3	y_4	y_5	$h_3(.)$
1	1	1	1	2
2	1	1	1	2
1	2	1	1	2
2	2	1	1	3
1	1	2	1	2
2	1	2	1	2
1	2	2	1	2
2	2	2	1	3
1	1	1	2	2
2	1	1	2	2
1	2	1	2	2
2	2	1	2	4
1	1	2	2	2
2	1	2	2	2
1	2	2	2	2
2	2	2	2	6

y_1	y_3	$h_4(.)$
1	1	14
2	1	8
1	2	7
2	2	7

d) $h_4(y_2,y_4)$

e) $h_3(y_1,y_3,y_4,y_5)$

Figure 3. The h functions

$h^r(y_3,y_4,y_5)$

y_3	y_4	y_5	$h^r(.)$	y^*_1
1	1	1	10	z_2
2	1	1	9	z_1
1	2	1	10	z_2
2	2	1	9	z_1
1	1	2	10	z_2
2	1	2	9	z_1
1	2	2	10	z_2
2	2	2	9	z_1

Figure 4. Step 2 - Iteration 1.

$h^r(y_2,y_4,y_5)$

y_2	y_4	y_5	$h^r(.)$	y^*_3	y^*_1
1	1	1	20	z_2	z_1
2	1	1	19	z_2	z_1
1	2	1	20	z_1	z_2
2	2	1	20	z_1	z_2
1	1	2	20	z_2	z_1
2	1	2	19	z_2	z_1
1	2	2	20	z_1	z_2
2	2	2	20	z_1	z_2

Figure 5. Step 2-Iteration 2.

y_2	y_4	y_5	$g(h^r,h_2)$	y^*_3	y^*_1
1	1	1	22	z_2	z_1
2	1	1	21	z_2	z_1
1	2	1	26	z_1	z_2
2	2	1	30	z_1	z_2
1	1	2	22	z_2	z_1
2	1	2	21	z_2	z_1
1	2	2	26	z_1	z_2
2	2	2	30	z_1	z_2

Figure 6. Final Step.