Locating Discretionary Service Facilities II: Maximizing Market Size, Minimizing Inconvenience

O. Berman, D. Bertsimas, R. C. Larson

OR 271-92 November 1992

Locating Discretionary Service Facilities II: Maximizing Market Size, Minimizing Inconvenience

by

Oded Berman*

Dimitris Bertsimas**

Richard C. Larson**

Revised November 1992

* University of Toronto, Division of Management, (Economics at Scarborough), The Faculty of Management, Toronto, Ontario, M5S 1V4, Canada

** Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139

ABSTRACT

Discretionary service facilities are providers of products and/or services that are purchased by customers who are traveling on otherwise pre-planned trips such as the daily commute. Optimum location of such facilities requires them to be at or near points in the transportation network having sizable flows of different potential customers. In [1] a first version of this problem was formulated, assuming that customers would make no deviations, no matter how small, from the pre-planned route to visit a discretionary service facility. Here the model is generalized in a number of directions, all sharing the property that the customer may deviate from the pre-planned route to visit a discretionary service facility. Three different generalizations are offered, two of which can be solved approximately by greedy heuristics and the third by any approximate or exact method used to solve the p-median problem. It is shown for those formulations yielding to a greedy heuristic approximate solution, including the formulation in [1], that the problems are examples of optimizing submodular functions for which the Nemhauser et. al. [5] bound on the performance of a greedy algorithm holds. In particular, the greedy solution is always within 37% of optimal, and for one of the formulations it is proved that the bound is tight.

In [1] Berman, Larson and Fouska introduced a new problem formulation in location theory which was called "optimal location of discretionary service facilities." The motivation was a perceived behavioral change on the part of customers. Instead of undertaking a one-stop tour from home or workplace to a facility to purchase a service or product, it was argued that many customers now carry out such purchases as part of routine pre-planned trips, say on the daily commute to and from home and workplace. Examples include stopping at gasoline service stations, automatic teller machines and "convenience stores." Traditional "Hakimi type" location models focus on minimizing some measure of travel distance or travel time from home (or workplace) to the facility. The optimal location of discretionary service facilities, on the other hand, requires convenience with regard to the customer's pre-planned trip.

The focus in [1] was on locating the m discretionary service facilities so as to maximize the flow of potential customers who passed at least one discretionary service facility along her preselected travel path from origin to destination. A path containing a facility was "covered;" a path not containing a facility, even if there existed a facility δ travel units from the path ($\delta > 0$), was not covered. In [1] it was proved that an optimal set of facility locations exists on the nodes of the network, and both exact and heuristic algorithms were developed to solve the problem.

In this paper we relax the assumption that, to be useful to a potential customer, a facility must be located at some point precisely on her pre-planned travel path. Facilities located "near" the pre-planned path may also be utilized by the customer.

In the first generalization, which we call "*delta coverage*" or problem (P_1), we assume that, as in [1], the customer passes through each of the nodes of her pre-planned trip path. If there are no facilities on the path, she is willing to detour a maximum distance Δ from any one of the path nodes to travel to a discretionary service facility. After purchasing the service or product at the facility, she returns to the same path node, implying that a total detour travel distance of up to 2Δ is incurred. This model depicts a situation in which a detour of up to 2Δ travel units, starting and ending at one of the pre-planned path nodes, is associated with a zero disutility on the part of the customer. Any detour requiring more than 2Δ travel units has in effect infinite disutility, so the detour will not be executed and the associated service (product) will not be purchased. We show that this problem can be reduced to the problem solved in [1].

In the second generalization, which we call "*maximize market size*" or (P_2), we assume that customers are increasingly likely to balk at traveling to a service facility as the deviation distance to it increases. More precisely, we assume that the probability that a customer is willing to travel an extra d units of distance to a facility that is located off the pre-planned path, assuming that this facility is the least inconvenient to the original path, is a convex decreasing function of d. The objective is to locate the m facilities so as to maximize the expected number of potential customers who become actual customers at the facilities. We develop both exact and approximate algorithms to solve this problem.

In the third generalization, which we call "*minimize expected inconvenience*" or (P_3), we assume that all potential customers traveling on the network must purchase the service at a service facility, regardless of the extra distance that must be traveled to get to the facility. In this sense the facilities are no longer discretionary. For some (lucky) customers, there will be a facility on their pre-planned travel paths, and no inconvenience is incurred. For others, the customers must deviate from the pre-planned trip to travel to the service facility causing least digression from the originally selected path. We assume that customers select their deviation paths to be the shortest ones possible. The objective of this problem is to locate the service facilities so as to minimize the total deviation distance traveled per unit of time, or equivalently, to minimize the expected deviation distance traveled by a random customer. We show that this problem is essentially an m median problem, and any m median algorithm can be used to locate the m facilities.

We present a generic worst case analysis of all the (greedy) heuristics developed both in this paper and in [1]. We show that each of our models, with the exception of the median model [problem (P_3)], belongs to a family of problems in which there is a "e^{-1}" worst case bound associated with the greedy heuristic and the bound is tight. This result improves upon the worst case bound published in [1] for the original form of the discretionary services location problem.

The paper concludes with specific algorithms, a greedy heuristic and a branch and bound algorithm, to solve problem (P_2). Numerical results are included.

1. Background and Notation

Let $G(N, A)$ be a bidirectional urban transportation network where N is the set of nodes with cardinality n and A is the set of arcs. We denote by $\underline{P}$ the set of non-zero flow paths through the network nodes and let f_p indicate the number of units of travel flow along any path $p \in \underline{P}$, per unit of time. Let m be the number of facilities to be located on the network. All facilities are assumed to provide identical service and thus no customer needs on any given trip to stop at more than one of them.

1.1. The Case of No Allowed Deviation

In [1] Berman, Larson and Fouska examined the problem [called problem (*BLF*)] of finding a set of m facilities on the network so as to maximize the total flow of different customers intercepted by the facilities under a very specific assumption regarding the behavior of customers. It was assumed that a customer may receive service only from facilities located on his (her) pre-planned trip path. In other words customers cannot deviate from their pre-planned paths. The problem can be formulated as

$$(BLF) \qquad \max_{\bar{x} \in G} \sum_{p \in \underline{P}} f_p I(\bar{x}, p)$$

where $I(\bar{x},p)$ is an indicator variable,

$$I(\bar{x}, p) = \begin{cases} 1 & \text{at least one } x \in \bar{x} \text{ is on path } p \\ 0 & \text{Otherwise} \end{cases}$$

and $\bar{x}$ is a vector of m points in G. In [1], it was shown that G can be replaced with N in problem (*BLF*) since it is proved that an optimal set locations exists in N.

1.2. Deviation Distances

In this paper we relax the assumption that customers do not deviate from their pre-planned trips when service is required. We define the deviation distance as the extra distance incurred when a customer deviates from his (her) pre-planned trip path. We denote by $d(a, b)$ the shortest distance (travel time) between a and b, $a, b \in G$.

To calculate the deviation distance for a customer who travels on path $p \in \underline{P}$, defined by a node visitation sequence $p = \{n_1, n_2, \ldots, n_l\}$ (where n_1 is the path p origin node and n_l is its destination node), we distinguish between two cases: Case *i*: $p \in \underline{P}$ is a shortest path; Case *ii*: $p \in \underline{P}$ is not a shortest path. When p is a shortest path, the deviation distance from the path p to the "nearest" of the m facilities, $D(p, \overline{x})$, is given by

$$D(p,\overline{x}) \equiv \min_{x \in \overline{x}} D(p, x) = \min_{x \in \overline{x}} \{d(n_j, x) + d(x, n_j) - d(n_1, n_l)\}.$$

When p is not a shortest path, we assume that the "path p" customer when traveling between origin n_1 and destination n_l must visit in proper sequence all the nodes contained in path p. Therefore, $D(p, \overline{x})$ is now given by

$$D(p,\overline{x}) \equiv \min_{n_j \in p, j \neq l} \left[\min_{x \in \overline{x}} \{d(n_j, x) + d(x, n_{j+1}) - d(n_j, n_{j+1})\}\right].$$

As an example for the calculations of $D(p, \overline{x})$ we refer to Figure 1 that depicts a simple network with 7 nodes. Suppose for Case *i*, that $p = (2, 1, 4, 5)$ and a single facility is located at node 3, then

$$D(p, \overline{x}) = 2 + 9 - 9 = 2$$

INSERT FIGURE 1 HERE.

For Case *ii*, let us assume that $p = (1, 2, 3, 6)$ and $\bar{x} = 7$, then $D(p, \bar{x}) = min\ \{6 + 5 - 3, 5 + 3 - 2, 3 + 5 - 3\} = 5$.

2. The Problems

2.1. (P_1): Delta Coverage

"Delta coverage" depicts a situation in which a detour of up to 2Δ travel units, starting *and* ending at the same pre-planned path node, is allowed in order for the customer to visit a facility "nearest" to her pre-planned route. It is assumed that the detour route is restricted to a tour comprising a minimum distance path from the detour-originating path node to a nearest facility and, due to network bidirectionality, a repeat of that path in reverse direction.

The formulation of problem (*BLF*) can be easily extended to include problem (P_1). In (P_1) a customer is said to be intercepted by a facility if at least one facility is at a distance of at most Δ from a node on the customer's trip path p, i.e.,

$$(P_1) \qquad \max_{\bar{x} \in G} \sum_{p \in \underline{P}} f_p I'(\bar{x}, p)$$

where we define

$$I'(\bar{x}, p) = \begin{cases} 1 & \exists\, j \in p \text{ such that } d(j, \bar{x}) \leq \Delta \\ 0 & \text{otherwise}, \end{cases}$$

where $d(j, \bar{x}) \equiv$ shortest distance between $j \in N$ and a nearest facility located at $x \in \bar{x}$.

Let us define N' as the union of the node set N and the set of all points N_Δ in G that are exactly Δ units of distance away from a node, i.e., $N' = N \cup N_\Delta$ where

$N_\Delta = \{y \in G \mid d(j, y) = \Delta, j \in N\}$. [Note that problem (P_1) reduces to problem (*BLF*) when $\Delta = 0$.]

Theorem 1. An optimal set of locations for problem (P_1) exists in N'.

Proof. Straightforward and omitted.

The algorithms of [1], both exact and heuristic, can be applied directly to solve problem (P_1), with the set of possible facility locations now extended to N'.

2.2. (P_2): Maximize Market Size

The objective of problem (P_2) is to locate the m facilities so as to maximize the expected number of potential customers who become actual customers at the facilities. Here we allow customers to deviate from their pre-planned route in the general manner described in Sec. 1.2; in particular, detours are not restricted to be tours. We assume that as the deviation distance grows larger, customers become less and less likely to select the detour to visit a nearest facility. Thus we again have a flow intercept maximization problem.

We assume that the flow of "path p" customers to the "nearest" of m facilities in the location set $\bar{x}$ is a convex decreasing function of the deviation distance $D(p, \bar{x})$, denoted by $f_p\, g(D(p, \bar{x}))$, where $g(0) \equiv 1$. Here $g(D(p, \bar{x}))$ can be interpreted to be the fraction of "path p" customers who would deviate to use a facility in $\bar{x}$ closest to path p, or equivalently, the probability that a random path p customer will deviate to use that facility. Therefore problem (P_2) is

$$(P_2) \qquad \max_{\bar{x} \in G} \sum_{p \in \underline{P}} f_p\, g(D(p, \bar{x})).$$

Theorem 2. An optimal set of locations to (P_2) exists in N.

Proof. *Case a* (all customer paths are minimum distance paths):

Path/facility assignment fixed: For any $x \in \bar{x}$, let $\underline{P}_x$ be the set of all paths in $\underline{P}$ that route all or a fraction of its customers to the facility at x. If x lies on path $p \in \underline{P}_x$, there is no deviation distance and the corresponding component of the objective function is f_p. Consider a path $p \in \underline{P}_x$ not containing x, where as usual n_1 is the origin node and n_l is the destination node of p. Suppose $x \in \bar{x}$ is an interior point on link (a, b), having length $d(a,b)$. Assume that the distance from node a to x is $Q\, d(a, b)$ where $0 \leq Q \leq 1$. Then the deviation distance from p to x, defined as $D(p, x)$, is found by computing the minimum of four possible detour routes to and from x:

$$D(p, x) = d(n_1, x) + d(x, n_l) - d(n_1, n_l) =$$

$$\min\{d(n_1, a) + 2Qd(a, b) + d(a, n_l) - d(n_1, n_l);\ d(n_1, a) + d(a, b) + d(b, n_l) - d(n_1, n_l);\ d(n_1, b) + d(a, b) + d(a, n_l) - d(n_1, n_l);\ d(n_1, b) + 2(1 - Q)d(a, b) + d(b, n_l) - d(n_1, n_l)\}.$$

Since $D(p, x)$ is the lower envelope of linear functions of Q, it is a piecewise linear concave function of Q. Since $g(y)$ is a decreasing convex function of y and $D(p, x)$ is a piecewise linear concave function of Q, $g(D(p, x))$ is a convex function of Q. Since a sum of convex functions is convex, we know that the partial sum

$$\sum_{p \in \underline{P}_x} f_p g(D(p, x))$$

is convex. Since Q is defined on the closed interval $0 \leq Q \leq 1$, a maximum of the partial sum must exist at an endpoint corresponding to $Q = 0$ or $Q = 1$.

Path/facility reassignment: Any change in location of the examined facility from an interior point on (a, b) to one of the arc's nodes may in turn cause a reassignment of customers on one or more paths in $\underline{P}_x$ to other facilities and/or may cause reassignment of customers on paths not in $\underline{P}_x$ to the examined facility. But such path/facility reassignment cannot decrease the objective function.

Case b (not all customer paths are minimum distance paths): Proved in a similar fashion. Q.E.D.

2.3. (P_3): Minimize Expected Inconvenience

In problem (P_3) *all* customers must travel to a service facility "closest" to their pre-planned paths to purchase or consume the service provided there. "Closeness" of a facility to a path is measured in terms of the minimum deviation distance (Sec. 1.2). The objective of problem (P_3) is to locate the service facilities so as to minimize the total deviation distance traveled per unit time, or equivalently, to minimize the expected deviation distance traveled by a random customer, i.e.,

$$(P_3) \qquad \min_{\bar{x} \in G} \sum_{p \in \underline{P}} f_p D(p,\bar{x})$$

The objective function of problem (P_3) is identical in form to that of the well known "m-median" type problem [2]. For the m-median problem, Hakimi proved that an optimal set of facility location exists on the nodes of the network [3]. Thus in (P_3) the search for optimal locations in G in the objective function can be replaced with a search limited to the node set N. Any of the algorithms, heuristic or exact, developed and used to solve the (NP-hard) m-median problem can be used for problem (P_3).

3. Worst Case Analysis of Greedy Algorithms

In this section we analyze a generic greedy algorithm for problems (P_1) and (P_2). We show that the greedy algorithm is always within 37% of the optimal solution and this bound is tight. In Section 4 we develop a specific greedy algorithm for problem (P_2), as well as a branch and bound algorithm, and we give numerical results.

Problems ($P1$) and ($P2$) can generically be formulated as follows: Let f_i: $2^N \rightarrow R+$ $(i = 1, 2)$ be set functions defined on subsets of the set N. Then, problem (P_i), ($i = 1, 2$) can be formulated

$$(P_i): \qquad Z_i = \max_{S \subseteq N_i, |S| \leq m} f_i(S)$$

where the functions $f_i(S)$ are defined in Section 2 and $N_1 = N' = N \cup N_\Delta, N_2 = N$.

Problem ($P3$), the median problem, can be formulated as

$$(P_3): \qquad \min_{S \subseteq N, |S| \leq m} f_i(S)\ .$$

Our focus in on problems ($P1$), ($P2$). A generic greedy algorithm for problems ($P1$), ($P2$) is:

<u>**Greedy Algorithm**</u>: *[Input: $f_i(S)$, m, N_i]*

[Output: Q_G, Z_G]

1. **(Initialization)** $Q^{\circ} \leftarrow \varnothing, t \leftarrow 1.$

2. **(Main Loop)** For $t = 1, \ldots, m$

   $j_t \leftarrow \operatorname{argmax}_{j \in N_i \setminus Q^{t-1}} f_i(Q^{t-1} \cup \{j\})$

   $Q^t \leftarrow Q^{t-1} \cup \{j_t\}$

3. **(Output)** $Q_G = Q^m$

$$Z_G = f(Q^m)$$

Given the function $f(S)$, the number of facilities m and the set of potential locations N_i, the algorithm outputs a set Q_G of m facilities with value Z_G.

Nemhauser, Wolsey and Fisher (1978), Fisher, Nemhauser and Wolsey (1978), and Nemhauser and Wolsey (1981) have studied the problem of

$\max_{S \subseteq N, |S| \leq m} f(S)$, where $f(S)$ is a submodular and nondecreasing function and where we have suppressed the subscript on the node set N_i and f_i. A set function is called submodular if for all $S, T \subseteq N$,

$$f(S \cap T) + f(S \cup T) \leq f(S) + f(T)$$

and is nondecreasing if for all $S, T,\ S \subseteq T$, we have $f(S) \leq f(T)$.

Obtaining the exact solution to the problem of maximizing a submodular set function is *NP*-hard. The greedy algorithm, however, approximates the problem very well, yielding results "close" to the optimum. More precisely, Nemhauser *et. al.* (1978) prove that

Theorem 3. (Nemhauser et. al. (1978))

The value Z_G returned by the greedy algorithm when applied to the problem:

$$Z_* = \max_{S \subseteq N, |S| \leq N} f(S) \qquad (1)$$

for $f(S)$ being submodular and nondecreasing satisfies

$$\frac{Z_G}{Z_*} \geq 1 - (1 - \frac{1}{m})^m \geq 1 - \frac{1}{e} \cong 0.63.$$

In other words the greedy algorithm returns a solution that is optimal for $m = 1$ and within 37% from the optimal solution value for any value of m. Moreover, the bound is tight, i.e., there are instances in which $Z_G = (Z_*)[1 - (1 - \frac{1}{m})^m]$.

Furthermore, Nemhauser and Wolsey (1979) have shown that within a large class of algorithms the greedy algorithm is the best possible for problem (1).

We plan to show in the remaining of this section that problems (P_1), (P_2) are instances of problem (1), i.e., the functions $f_i(s)$, $(i = 1, 2)$ are submodular and nondecreasing. We will also show that problem (P_3) is an instance of minimizing a supermodular nonincreasing function.

3.1 Delta Coverage

With the usual bidirectional network $G = (N, A)$, let $d(x, i)$ be the length of the shortest path from node x to node i $(x, i \in N)$ and let $N(x, \Delta) = \{ i \in N: d(x, i) \leq \Delta\}$ be the set of nodal "delta coverage" points associated with node x. Let $\underline{P}$ be the given set of paths, assuming that a path $p \in \underline{P}$ is specified as a set of nodes in N, i.e., $p = \{n_1, \ldots, n_l\}$. A customer whose path includes node x could travel from x to a service facility located in $N(x, \Delta)$, and back, incurring a detour travel distance not greater than 2Δ; in that way any facility in $N(x, \Delta)$ "covers" node x. Or, equivalently, a customer whose travel path includes at least one node n_j in $N(x, \Delta)$ could travel from that node to a service facility located at x and back, incurring a detour travel distance not exceeding 2Δ; in that way a facility located at x covers any node $n_j \in N(x, \Delta)$.

Let

$$N(S, \Delta) = \bigcup_{x \in S} N(x, \Delta)$$

be the set of nodal "delta coverage" points corresponding to any subset S of potential facility locations, $S \subseteq N \cup N_\Delta$ (see Figure 2). A customer whose travel path includes at least one node in $N(S, \Delta)$ could travel from that path node to and from a facility, with a detour travel distance not exceeding 2Δ, if at least one facility is located in S.

INSERT FIGURE 2 HERE

Using Theorem 1, problem $(P1)$ is formulated as

$$(P1) \qquad \underset{\substack{S \subseteq N \cup N_\Delta \\ |S| \leq m}}{Max} \qquad f_1(S) = \sum_{\substack{p \in \underline{P} \\ p \cap N(S, \Delta) \neq \emptyset}} f_p$$

Note that for $\Delta = 0$, $N(x, \Delta) = \{x\}$, $N(S, \Delta) = S$ and the problem reduces to the problem studied in Berman *et. al.* (1992).

Proposition 1

If $f_p \geq 0$ for all $p \in \underline{P}$, then for any $\Delta \geq 0$, $f_1(S)$ is submodular and nondecreasing.

Proof:

- If $S \subseteq T$, then clearly $N(S, \Delta) \subseteq N(T, \Delta)$ which implies that $f_1(S) \leq f_1(T)$ if $f_p \geq 0$, i.e., $f_1(S)$ is nondecreasing.

- In order to show that a set function is submodular it suffices to show that for *all* $S \subseteq T$ and $k \notin T$, $f_1(T \cup \{k\}) - f_1(T) \leq f_1(S \cup k) - f_1(S)$.

For any S and T, $S \subseteq T$, let $\underline{P}_T = \{p \in \underline{P}: p \cap N(T, \Delta) = \varnothing\}$, i.e., $\underline{P}_T$ is the set of all paths not covered by locating facilities in T. Since $S \subseteq T$, then $N(S, \Delta) \subseteq N(T, \Delta)$ which implies that $\underline{P}_T \subseteq \underline{P}_S$. Then,

$$f_1(T \cup \{k\}) - f_1(T) = \sum_{\substack{p \in \underline{P} \\ p \cap N(T \cup \{k\}, \Delta) \neq \varnothing}} f_p - \sum_{\substack{p \in \underline{P} \\ p \cap N(T, \Delta) \neq \varnothing}} f_p$$

$$= \sum_{\substack{p \in \underline{P} \\ p \cap N(T, \Delta) = \varnothing \\ p \cap N(k, \Delta) \neq \varnothing}} f_p = \sum_{\substack{p \in \underline{P}_T \\ p \cap N(k, \Delta) \neq \varnothing}} f_p$$

$$\leq \sum_{\substack{p \in \underline{P}_S \\ p \cap N(k, \Delta) \neq \varnothing}} f_p$$

$$= f_1(S \cup \{k\}) - f_1(S),$$

i.e., $f_1(S)$ is submodular.

3.2. Maximize Market Size

In Sec. 1.2 path deviation distances were defined for two cases: (1) all customer paths are minimum distance paths; (2) not all customer paths are minimum distance paths. For the former case, for each $p \in \underline{P}$ and $S \subseteq N$, we can write the deviation distance of path p from set S as

$$D(p, S) = \min_{\substack{i, j \in p \\ x \in S}} [\, d(i, x) + d(x, j) - d(i, j)\}.$$

Let $g(y) : R^+ \rightarrow R^+$ be a nondecreasing function not necessarily convex. From Theorem 2, Problem (P_2) can be formulated as

$$(P_2): \quad \underset{\substack{S \subseteq N \\ |S| \leq m}}{Max} \; f_2(S) = \sum_{p \in \underline{P}} f_p \; g(D(p, S)).$$

In the following proof, we assume that all customer paths are minimum distance paths. A directly analogous proof applies to the other case.

Proposition 2

If $g(y)$ is a nonincreasing function, then $f_2(S)$ is a submodular, nondecreasing set function. On the other hand, $f_3(S)$ a is supermodular, nonincreasing set function.

Proof.

If $S \subseteq T$, then $D(p, S) \geq D(p, T)$ which implies that $g(D(p, S)) \leq g(D(p, T))$, since $g(y)$ is nonincreasing. Therefore, $f_2(S) \leq f_2(T)$, i.e., $f_2(S)$ is nondecreasing. Similarly, $f_3(S) \geq f_3\,(T)$, i.e., $f_3(S)$ is nonincreasing.

We will now show that $g(D(p, S))$ is submodular. Let $S \subseteq T$ and $k \notin T$. We first note that $D(p, T \cup \{k\})] = min\,\{D(p, T), \underset{i, j \in p}{Min}\,[d(i, k) + d(k, j) - d(i, j)]\}$.

In order to check whether $g(D(p, S))$ is submodular it suffices to show that

$$g(D(p, T \cup \{k\})) - g(D(p, T)) \leq g(D(p, S \cup \{k\})) - g(D(p, S)),$$

or equivalently, defining $\alpha \equiv \underset{i, j \in p}{Min}\,[d(i, k) + d(k, j) - d(i, j)]$,

$$g(Min[D(p, T), \alpha]) - g(D(p, T)) \leq g(Min\,[D(p, S), \alpha]) - g(D(p, S)). \qquad (2)$$

In order to check *(2)* we distinguish three cases:

1. $\alpha \geq D(p, S) \geq D(p, T)$.

   Then *(2)* becomes

$$g(D(p, T)) - g(D(p, T)) \leq g(D(p, S)) - g(D,(p, S)),$$

   which is obviously satisfied.

2. $D(p, S) \geq \alpha \geq D(p, T)$

   Then *(2)* becomes

$$0 \leq g(\alpha) - g(D(p, S)),$$

   which is satisfied since $g(y)$ is nondecreasing.

3. $D(p, S) \geq d(p, T) \geq \alpha$

   Then *(2)* becomes

$$g(\alpha) - g(D(p, T)) \leq g(\alpha) - g(D(p, S)),$$

   which is satisfied since $g(D(p, S)) \leq g(D(p, T)$. ($S \subseteq T$ and $D(p, S) \geq D(p, T)$).

Therefore the set function $g(D(p, S))$ is submodular. Then, if $f_p \geq 0, f_2(S)$ is also submodular since it is a sum of submodular set functions.

By exactly the same proof we can show that $D(p, S)$ is supermodular, i.e., $D(p, T \cup \{k\}) - D(p, T) \geq D(p, S \cup \{k\}) - D(p, S)$ which implies that $f_3(S)$ is supermodular. Note that the submodularity of $f2(S)$ is independent of any convexity assumption on $g(x)$.

3.3 Main Theorem

Combining Theorem 3 and Propositions 1 and 2, we can prove

Theorem 4. For problems (P_1), (P_2) the generic greedy algorithm produces a value Z_{G_i} $(i = 1, 2)$ such that

$$\frac{Z_{G_i}}{Z_{*i}} \geq 1 - (1 - \frac{1}{m})^m. \tag{3}$$

In the following theorem we investigate the tightness of the bound.

Theorem 5. For problem (P_1) with $\Delta = 0$, the bound (3) is tight.

Proof: Consider a network with nodes $A_0, A_1, \ldots, A_m, B_1, B_2, \ldots, B_{m-2}, C_1, \ldots, C_m$. The set of edges is as follows: $E = \{(A_0, A_i), (A_i, C_i); i=1, \ldots, m\} \cup \{(A_i, B_j); i = 1, \ldots, m, j=1, \ldots, m-2\}$. (See Figure 3 for m=4.) The set $\underline{P}$ of paths is as follows: For $i = 1, \ldots m$

m paths (A_0, A_i) each with value $f_0 = m^{m-2}$

m paths (A_i, B_1) each with value $f_1 = \frac{m-1}{m} m^{m-2}$

.
.
.

m paths (A_i, B_j) each with value $f_j = \left(\frac{m-1}{m}\right)^j m^{m-2}$, j = 1 . . . m-2

m paths (A_i, C_i) each with value $f_{m-1} = (m-1)^{m-1}$

INSERT FIGURE 3 HERE

The optimal solution of problem (P_1) with $\Delta = 0$ and up to m facilities is the set $(A_1, \ldots, A_m)$ of the nodes at the first level that covers all the paths. The value of the optimal solution is

$$Z_* = mf_0 + m \sum_{j=1}^{m-2} f_j + mf_{m-1} = m^m$$

The greedy algorithm selects node A_0 first, then node B_1, then node $B_2, \ldots,$ node B_{m-2} and finally one of $A_1, \ldots, A_m$.

The cost of the greedy algorithm is

$$Z_G = mf_0 + m \sum_{j=1}^{m-2} f_j + f_{m-1} = m^m - (m-1)^m.$$

The reason is that the value of A_0 is at least as large as any other's node, B_1 is at least as high as any from the remaining nodes, etc. Thus, $\frac{Z_G}{Z_*} = 1 - (1 - \frac{1}{m})^m$. *Q. E. D.*

Problem *(P3)* is a minimization of a supermodular nonincreasing function subject to a cardinality constraint. Although approximating a maximization of a submodular function within a constant factor is possible as we have seen, approximating a minimization of a supermodular function within a constant is *NP*-hard. Although the supermodular function $f_3(S)$ appearing in problem *(P3)* has special structure, we strongly suspect that the problem does not have a constant guaranteed heuristic. Indeed the greedy algorithm which was very successful for problems *(P1)* and *(P2)* delivers a solution with cost arbitrarily far from the optimal cost.

4. Solving Problem P_2

In this section we focus on solution methods for problem (P_2). Building on our standard notation, let $D(p, i)$ be the deviation distance for a path p customer to detour through node i. This notation applies whether or not strictly minimum distance paths are used. Define the *potential* expected path p flow through node i as

$$C_{pi} = f_p g(D(p,i)) \equiv f_p g(\min_{j \in p} D(j,i)).$$

Since $g(.)$ is a convex decreasing function,

$$\sum_{p \in \underline{P}} f_p\, g(D(\bar{x}, p)) = \sum_{p \in \underline{P}} \max_{i \in \bar{x}} C_{pi}$$

Therefore problem (P_2) could be formulated as

$$(P_2) \qquad \max_{\bar{x} \in N} \sum_{p} \max_{i \in \bar{x}} C_{pi}$$

We can now identify binary decision variables and assignment variables. Let

$$X_{pi} = \begin{cases} 1 & \text{if a facility i is assigned to path } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$X_j = \begin{cases} 1 & \text{if a facility is located at node j} \\ 0 & \text{otherwise} \end{cases}$$

Now problem (P_2) can be stated as a mathematical programming problem:

$$(P_2) \qquad \max \sum_{p \in P} \sum_{i=1}^{n} C_{pi} X_{pi}$$

s.t.

$$\sum_{j=1}^{n} X_j = m$$

$$X_j - X_{pj} \geq 0 \qquad j = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} X_{pj} = 1 \qquad \text{for all } p \in \underline{P}$$

The first constraint guarantees that *m* facilities are located. In the second set of constraints we have made sure that path *p* cannot be assigned to a node that does not house a facility, whereas the last set of constraints ensures that each path is assigned to exactly one facility.

To solve problem (P_2), we developed a greedy heuristic (having the properties of the generic greedy heuristic of Sec. 3) and an exact algorithm. The greedy heuristic is a modification of the greedy heuristic to solve problem *(BLF)* (See [1].).

<u>The Greedy Heuristic to Solve P_2</u>

Step 1. $l = 1$

Step 2 Compute $b_i = \Sigma_{p \in \underline{P}} f_p g(d(i,p)), \quad i=1, \ldots, n.$

Step 3. Find $b_{i_{max}} = max_{i \in N} \{b_i\}$; locate facility l at node i_{max} and delete i_{max} from N.

Step 4. Set $\forall p \in \underline{P}, \forall j \in N,$

$$g(D(i, p)) = [g\, D(i, p)) - g(D\,(i_{max}\, p))]^+$$

Delete from $\underline{P}$ the set $\underline{P}_{i_{max}}$ (which is the set of all paths p for which $g(D(i_{max}, p)) = 0$.

Step 5. If $l = m$ or if $\underline{P} = 0$, STOP. Otherwise set $l = l + 1$ and go to *Step 2.*

As an example, let us refer again to the network in Figure 1 and let us solve problem (P_3) with an exponential customer damping factor, i.e., $g(D(p, \bar{x})) = e^{-bD(p,\bar{x})}$. The paths and all extra distances are given in Table 1. Suppose $b = .05$ and $f_p = 1$ $\forall p \in \underline{P}$. We start the heuristic with $l = 1$.

Since $b_1 = 9.413$; $b_2 = 8.781$; $b_3 = 8.993$; $b_4 = 9.065$; $b_5 = 8.171$; $b_6 = 9.4999$; $b_7 = 8.722$; $b_{jmax} = b_6$, we locate facility *1* at node *6* and we delete node *6* from *N*. We find new $e^{-bD(p,j)}$ $\forall p, j \neq 6$ and delete from $\underline{P}$ paths *4, 5, 6, 7, 8, 9*. Since $l = 1, \underline{P} = \emptyset$, we let $l = 2$. Now $b_1 = .0975$; $b_2 = 0.2300$; $b_3 = .1812$, $b_4 = .0975$; $b_5 = .3187$, $b_7 = .4024$, $b_{jmax} = b_7$ and we locate the second facility at node *7*, and we delete node *7* from *N*. We find new $e^{-bD(p,\ j)}$ and delete from $\underline{P}$ paths 2 and *10*. Since $l = 2, P = \emptyset$ we set $l = 3$, find $b_1 = 0.0975$; $b_2 = .0487$, $b_3 = 0$; $b_4 = .0975$; $b_5 = .0975$, $j_{max} = 1$ (or 4 or 5). Since $l = 3$, we are done. Total flow intercepted 9.4999 + 0.4024 + 0.0975 = 9.9998.

INSERT TABLE 1 HERE

To solve the problem (P_2) exactly we developed a branch and bound code for the problem. This branch and bound procedure is based on two upper bounds. The first one which is a minor modification of the one developed in [1] for (P_1) is called UB_1. The variables $X_1, \ldots, X_n$ are the decision variables for the branch and bound tree. Let us define $D \subset N$ as a set of all variables that constitute a partial solution in the branch and bound tree (i.e., $j \in D \Rightarrow X_j = 0, 1$) and let $U = N - D$. Let $D_1 \subset D$ $|D_1| = l$, be the set of all nodes in D that house a facility ($j \in D_1 \Rightarrow X_j = 1$) and let $D_0 = D - D_1$ ($j \in D_0 \Rightarrow X_j = 0$). Let

$$r_i = \max_{j \in D_1} C_{ij} \quad i = 1, \ldots, |\underline{P}|,$$

i.e., r_i is the maximum amount of flow that the facilities in the partial solution intercept from path i. For each $j \in U$ we define $\delta_j = \sum_{i=1}^{|\underline{P}|} \max\{0, C_{ij} - r_j\}$, i.e., δ_j is maximum amount of flow node j can intercept after deleting all the flow intercepted by the already-located facilities. Now UB_1 can be defined

$$UB_1 = \sum_{i=1}^{|\underline{P}|} r_i + L$$

where L is the sum of the $(m - l)$ greatest δ_j's. The second upper bound called UB_2 can be now defined

$$UB_2 = \sum_{p \in P} \max_{j \in D_1 \cup U} C_{pj}$$

We note that UB_2 is useful when there are many variables in a partial solution for which $X_j = 0$ which is exactly the situation when UB_1 is not useful. Therefore UB_1 and UB_2 complement each other in the branch and bound. Finally, the upper bound is the minimum of UB_1 and UB_2.

The computer code implementing the branch and bound algorithm for problem (P_3) is written in C and tests were run on DEC 5810. In order to provide test results we generated randomly network sizes, their path and corresponding flows. In Table 2, we illustrate a typical sample of our test cases for the problem with an exponential damping factor. The table provides the CPU time and the ratio of the solution value provided by the greedy heuristic and the branch and bound for networks with number of nodes and number of paths ranging from 20 to 100 and number of facilities ranging from 2 to 5. We see that for this set of runs the greedy heuristic performs considerably better than its worst case bound.

INSERT TABLE 2 HERE

Acknowledgments.

The research of Oded Berman was partially supported by the National Science and Engineering Council of Canada. The research of Dimitris Bertsimas was partially supported by the National Science Foundation under grant DDM-9014751 and by the Presidential Young Investigator award DDM-9158118 with matching funds from Draper Laboratory. The work of Richard Larson was supported by the National Science Foundation, Grant SES-9119962.

References

1. Berman, O., R. C. Larson and N. Fouska, "Optimal Location of Discretionary Service Facilities," *Transportation Science*, 26(23), 201-211, 1992.

2. Mirchandani, P. B. and R. L. Francis, *Discrete Location Theory*, John Wiley, New York, 1990.

3. Nemhauser, G. L. and L. A. Wolsey, "Best Algorithms for Approximating the Maximum of a Submodular Set Function," *Mathematics of Operations Research* 3, 177-188, 1981.

4. Nemhauser, G. L. and L. A. Wolsey, "Maximizing Submodular Set Functions: Formulations and Analysis of Algorithms," *Annals of Discrete Mathematics* 11, 279-301, 1981.

5. Nemhauser, G. L., L. A. Wolsey and M. L. Fisher, "An Analysis of Approximations for Maximizing Submodular Set Functions," *Mathematical Programming* 14, 265-294, 1978.

6. Hakimi, S. L., "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," *Operations Research* 12, 450-459, 1964.

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

Figures:

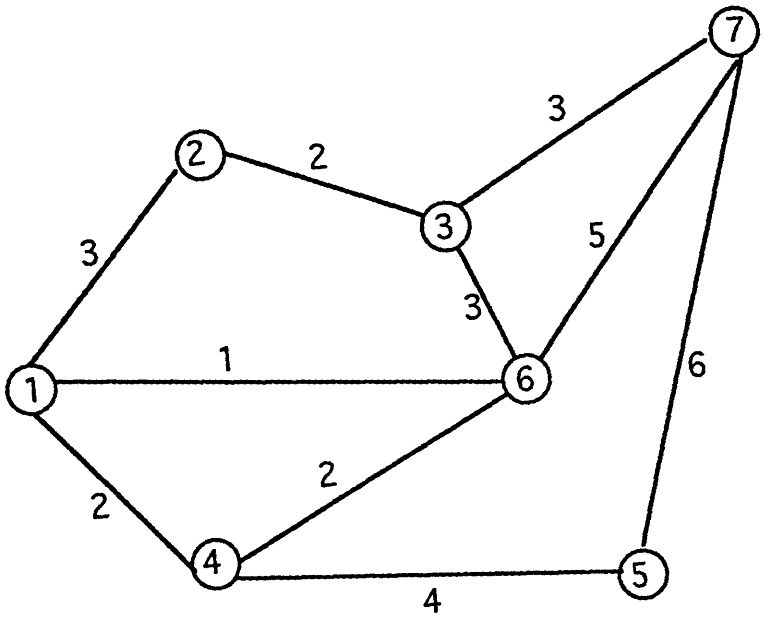
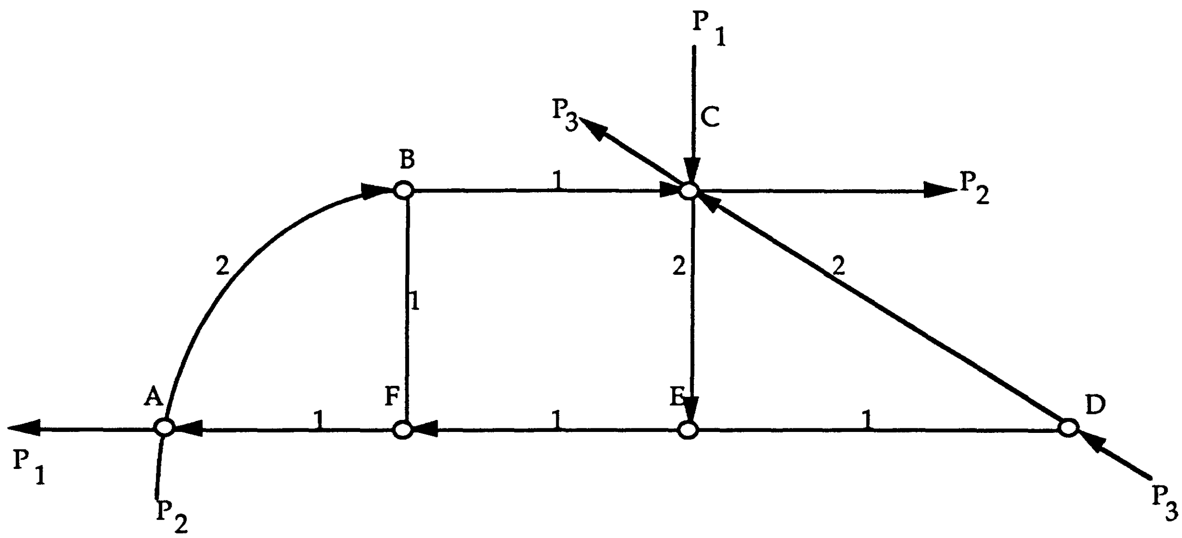


Figure 1

A 7-Node Network



$p_1 = \{C, E, F, A\}$ $\quad fp_1 = 2$

$p_2 = \{A, B, C\}$ $\quad fp_2 = 2$

$p_3 = \{D, C\}$ $\quad fp_3 = 1$

$\Delta = 1$

$S = \{A, F\}$

$N(A, 1) = \{A, F\}, N(F\ 1) = \{A, F, B, E\}$

$f_1(S) = fp_1 + fp_2 = 4$

Figure 2

Example for Delta Coverage

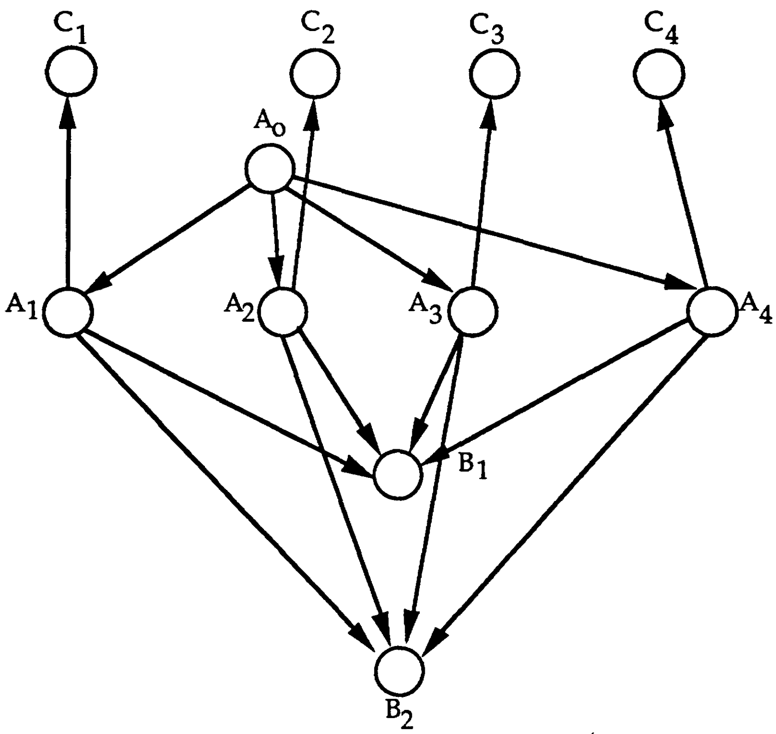


Figure 3. Example in the Proof for $m = 4$.

TABLES:

Nodes/Path		1	2	3	4	5	6	7
(1)	2145	0	0	2	0	0	1	2
(2)	237	4	0	0	7	10	4	0
(3)	145	0	6	7	0	0	1	6
(4)	167	0	2	1	3	6	0	0
(5)	361	0	1	0	3	11	0	5
(6)	3645	1	2	0	0	0	0	0
(7)	463	1	2	0	0	6	0	5
(8)	546	1	2	6	0	0	0	5
(9)	612	0	0	1	3	11	0	6
(10)	75	6	8	6	5	0	5	0

Table 1: Paths and Extra Distances for the Example.

n	$\|P\|$	m	CPU of the B&B Algorithm	Value of Greedy / Value of Branch and Bound
10,	10,	2	0	1
		3	0	1
		4	0	1
		5	0	1
30,	30	2	0	.998
		3	1	.937
		4	5	.951
		5	22	.993
50,	50	2	1	.983
		3	11	.935
		4	69	.968
		5	355	.955
100,	100	2	5	1
		3	169	.936
		4	693	.934
		5	2147	.943

Table 2

CPU Times in Seconds (rounded to closest integer) and ratio of objective function of the greedy heuristic to the objective function of the branch and bound procedure for several n, $|P|$, m values.