Locating flow-intercepting facilities: New approaches and results*

Oded Berman, Dmitry Krass and Chen Wei Xu
Faculty of Management, University of Toronto, 246 Bloor Street West, Toronto, Ontario, Canada M5S 1V4

The problem of locating flow-intercepting facilities on a network with probabilistic customer flows and with facility set-up costs is studied in this paper. Two types of models are investigated, namely the double-counting model and the no-double-counting model (double-counting refers to multiple interceptions of the same customer). For each model, a nonlinear integer programming formulation is first obtained via the theory of Markov chains, and an equivalent linear integer program is then derived. A simple greedy heuristic is proposed for solving both models and a worst-case bound is established, which is shown to be tight under certain conditions.

1. Introduction

The problem of locating a set of facilities on a network so as to intercept the maximum proportion of pre-existing customer flows appears in many applications ranging from locating traffic monitors and police radar units on traffic networks to finding optimal places for advertising vehicles to maximize exposure.

The basic underlying problem was analyzed in Berman et al. [3] and Hodgson [9], with extensions and generalizations appearing in Berman et al. [2] and Berman [1]. In these papers, the formulation of the model depends critically on the availability of the customer flow data for all paths of the network. This requirement is very restrictive due to both the large volume and the unavailability of the required data. Normally, only the data on the flow between all origin-destination (OD) pairs are available. The usual "work-around" is to assume that all flow for each OD pair occurs along the shortest path. However, this assumption is known to contradict the actual trip behavior (e.g., see Dial [8] for more discussion). Another way of obtaining path flow data from OD-flow is to solve a network traffic equilibrium problem (see,

*This work is in part supported by the NSERC grants of O. Berman and D. Krass.

e.g., Sheffi [14]). But this approach also suffers from two major shortcomings: (1) an explicit enumeration of all paths is essential to finding a solution, and (2) it is based on several strong assumptions such as: travellers have complete information on the traffic loads along the various travel paths.

This data problem is bypassed in the "probabilistic customer flow" approach, where we only require the traffic volumes for all OD pairs and the "turning probabilities", i.e., the probability that a customer arriving at an intermediate node will take a particular link out of that node. These turning probabilities can either be obtained by direct observation of the traffic flows, or estimated by several available methods – e.g., Dial's STOCH algorithm or Boeing's FLAIR (automatic vehicle monitoring) system (see Lewis and Leznick [11]). The formulation and analyses for the flow-intercepting location problem with probabilistic customer flow data are presented in Berman et al. [4,5], where efficient integer programming formulations for various versions of the model are developed via the theory of constrained Markov decision processes (MDP) [10].

The current paper contains several new results on the probabilistic approach. First, we present a general formulation of the model, which includes both fixed and variable costs for operating the facilities. Our model incorporates the cases in which only the first visit by a customer to a facility has any value (which normally holds for inspection-type applications), as well as the cases where secondary visits also have some value (as in, for example, advertising applications).

Second, we develop an integer programming (IP) formulation for each variant of the model directly from the elementary theory of Markov chains (see, e.g., Çinlar [6]). The approach used in the previous papers relies on the development of IP formulations via the theory of average- or total-reward MDPs with constraints on state-action frequencies, and thus is fairly challenging to follow. The approach presented in this paper is both more elegant and significantly more accessible.

Third, we develop a greedy heuristic for our models and show that, under certain conditions, its worst-case performance is quite impressive – the relative error is on the order of e^{-1}. Moreover, the worst-case bound can be shown to be tight. Finally, we present the results of some computational experiments which show that the actual performance of the heuristic significantly exceeds the worst-case bound in most randomly-generated cases, usually providing near-optimal solutions. Thus, excellent approximate solutions can usually be constructed for problems of very large dimensionality.

The paper is organized as follows. The basic model is described in section 2, where a formulation as a nonlinear Integer Program is developed. An equivalent Linear Integer Program is constructed in section 3. The greedy heuristic, its worst-case bound, and an example showing the tightness of the worst-case bound, are presented in section 4. The results of a set of computational experiments aimed at comparing the actual performance of the greedy heuristic to the optimal solutions under different conditions are presented in section 5.

2 Model formulation

Following Berman et al. [4,5], we consider the following basic problem. Given a network with node set $N = \{1,\dots,n\}$ and some pre-existing customer flows, we seek to find optimal nodal locations for a set of facilities so as to maximize the expected proportion of the total flow that is "intercepted" by all facilities. Intuitively, an incidence of "interception" occurs when a customer passes in a close vicinity of the facility and makes a conscious decision to either accept, or decline the service offered there. In our model, the customer is required to travel through a node containing a facility in order for an "interception" to occur. However, we will allow for the possibility that a customer needing service might nevertheless bypass a particular facility due to inconvenient on/off ramp, poor visibility of the facility, etc. This situation will be modeled as follows: for each node i on the network we define an "obscureness index" $\beta_i \in (0, 1)$, and we assume that if there is a facility at node i, a customer passing through this node has a $1 - \beta_i$ chance of being intercepted by the facility (thus, more attractive locations have a lower value of β_i). Note that a facility may only intercept the customers that have not previously been intercepted by other "upstream" facilities. This reflects the situation where customers require at most one instance of service during a single sojourn through the network (e.g., in the case of gas stations). In some other situations, a "multiple service" model might be more appropriate – see remark 2 at the end of this section for more discussion. The objective function considered in the previous work on this and related problems has been to maximize the expected fraction of the total customer flow intercepted by all facilities (see Berman et al. [2–5], Hodgson [9], Berman [1]). This objective is based on the assumption that the profit earned by the facilities is proportional to the customer flow intercepted by them; the fixed operating and set-up costs are (implicitly) assumed to be equal for all the facilities. The latter assumption is especially questionable in our model: since we explicitly account for the situation where some locations are more attractive than others, it is reasonable to assume that this would be reflected in the set-up costs. Thus, we will assume that each node i has an associated fixed cost s_i, which includes the set-up and fixed operating costs. Let f_i be the expected fraction of the total customer flow intercepted by a facility at node i. We will assume that the expected value of the revenue generated by this facility will be equal to rf_i (where r is the revenue that would be earned by intercepting all customers in the system). The overall objective is to maximize the expected profit, which is equal to the total expected revenue earned by all facilities on the network minus the fixed costs incurred in operating these facilities.

Remark 1

It should be noted that nodal optimality need not hold for generalized flow interception models (see Berman et al. [5]). This is in contrast to the case of the basic flow interception model introduced in Berman et al. [4]. Nevertheless, from a practical point of view the assumption of allowing only nodal locations for facilities is not unreasonable since usually a finite set of candidate facility sites is considered. □

Following Berman et al. [5], the model is formally defined as follows. For $i, j \in N$, let b_i be the fraction of customers starting their travel from node i, and let c_{ij} be the fraction of customers traveling to node j immediately after reaching node i (i.e., c_{ij} represents the conditional probability P(next node is j | current node is i)). We adjoin to the network an artificial "common sink" node 0, and set $b_0 = 0$, $c_{00} = 1$ and c_{i0} equal to the fraction of customers leaving the network (completing their travel) immediately after visiting node i. We also add another artificial sink node $(n+1)$, with $b_{n+1} = 0$, $c_{i(n+1)} = 0$ for $i \in N$ and $c_{(n+1)(n+1)} = 1$ (note that this node is not reachable from any other node on the network – it will be used to model the intercepted flow, as shown below). Thus the customer flow data can be summarized by an $(n+2) \times (n+2)$-dimensional Markov matrix C^+ and an $(n+2)$-dimensional vector $\boldsymbol{b}$ (we reserve the notation C to represent the $n \times n$ submatrix of C^+ corresponding to the nodes in N).

We make the following assumption:

ASSUMPTION 1

All customers complete their travel in finite time. Note that this condition is equivalent to the requirement that the only recurrent classes of C are $\{0\}$ and $\{n+1\}$.

The decision variables are u_i, $i \in N$, where $u_i = 1$ if a facility is placed at node i, and $u_i = 0$ otherwise. We will refer to the n-dimensional binary vector $\boldsymbol{u}$ as the "location vector". Every location vector $\boldsymbol{u}$ induces a new Markov transition matrix $C^+(\boldsymbol{u})$ with entries defined as follows:

$$
\begin{aligned}
c_{i,j}(\boldsymbol{u}) &= c_{ij}, && \text{if } u_i = 0,\ i \in N,\ j \in N \cup \{0, n+1\} \text{ or } i \in \{0, n+1\},\\
c_{i,j}(\boldsymbol{u}) &= \beta_i c_{ij}, && \text{if } u_i = 1,\ i \in N,\ j \in N \cup \{0\},\\
c_{i,n+1}(\boldsymbol{u}) &= 1 - \beta_i, && \text{if } u_i = 1,\ i \in N
\end{aligned}
$$

(intuitively, placing a facility at node i has the effect of diverting the incoming flow to node $(n+1)$ with probability $1 - \beta_i$). The model is pictorially represented in figure 1.

Let $f(\boldsymbol{u})$ be the probability of ever reaching node $(n+1)$ in the Markov chain induced by $C^+(\boldsymbol{u})$, with the initial distribution $\boldsymbol{b}$. Note that this is equal to the expected amount of flow intercepted by all facilities under the location vector $\boldsymbol{u}$.

The problem can now be formulated as

$$\max_{\boldsymbol{u}} \left[r f(\boldsymbol{u}) - \sum_{i \in N} s_i u_i \right].$$

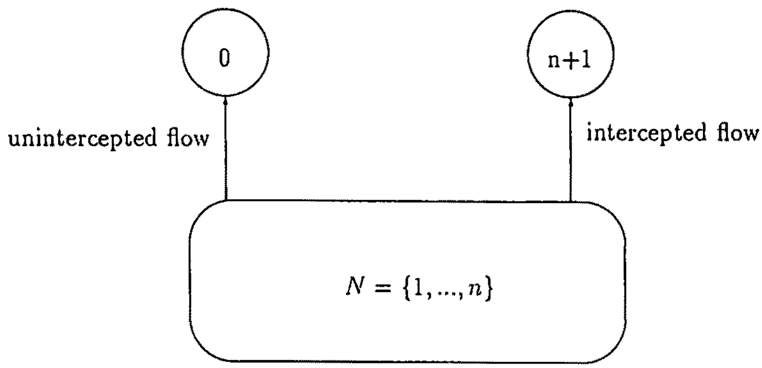

Figure 1. Illustration of the network structure.

This formulation can be written in terms of the original data by employing the standard Markov chain techniques. First we decompose the transition matrix $C^+(\boldsymbol{u})$ as follows:

$$C^+(\boldsymbol{u}) = \begin{bmatrix} 1, \ 0,\ldots,0 & 0 \\ C(\boldsymbol{u}) & \boldsymbol{c}_{n+1}(\boldsymbol{u}) \\ 0,\ldots,0 & 1 \end{bmatrix}.$$

Thus, $C(\boldsymbol{u})$ represents the $n \times n$-dimensional submatrix of $C^+(\boldsymbol{u})$ corresponding to the rows and columns in N, and $\boldsymbol{c}_{n+1}(\boldsymbol{u})$ is an n-dimensional column vector. From the elementary theory of Markov chains, it follows that

$$f(\boldsymbol{u}) = \boldsymbol{b}^T[I - C(\boldsymbol{u})]^{-1}\boldsymbol{c}_{n+1}(\boldsymbol{u}). \tag{1}$$

Using (1), we obtain the following Nonlinear Integer Program:

(NLP1)

$$\text{maximize} \quad r\boldsymbol{b}^T[I - C(\boldsymbol{u})]^{-1}\boldsymbol{c}_{n+1}(\boldsymbol{u}) - \sum_{i \in N} s_i u_i \tag{2}$$

$$\text{subject to} \quad u_i \in \{0,1\}, \quad i \in N. \tag{3}$$

The above formulation is obviously unwieldy and not easily solvable by any of the standard Integer Programming techniques. In section 3 below, we will develop a much more tractable formulation of this model as a Linear Integer Program. We close the current section with a few comments and remarks on the model described above.

Note that in (NLP1) there is no restriction on the number of facilities that may be located. This is typically not the case in practice, where budgetary and other constraints frequently limit the maximal number of new facilities that may be considered. To reflect this, we will assume that a value m is specified as a maximal number of new facilities that can be opened, and we will include the following additional constraint in the (NLP1) above:

$$\sum_{i \in N} u_i \leq m.$$

The probabilistic flow-capturing model described in this section is largely equivalent to model FLP(2, β) ("2" stands for model 2) described in Berman et al. [5], the major difference lying in the introduction of the set-up costs explicitly in the model and in changing the objective function to profit maximization from the maximization of the total expected intercepted flow (the number of facilities was required to be m in previous models).

Remark 2

As discussed in Berman et al. [5], the model developed above is mainly applicable in the situations where there is no possibility of a customer obtaining service more than once during a single sojourn on the network (i.e., the value of secondary exposures to facilities is 0). This is likely to be the case for many applications such as locating automatic teller machines, gas stations, police sobriety checkpoints, etc. However, there is a large class of other potential applications, mainly in marketing, where the value of secondary, and subsequent, exposures to a facility is likely to be quite substantial. This is the case, for example, in choosing optimal locations for advertising billboards or television commercials. Berman et al. [5] develop an alternative model, FLP(1, α) ("1" stands for model 1), which explicitly allows a customer to be "intercepted" by several facilities (the parameter α represents the value of a secondary exposure as a percentage of the value of the initial exposure). It is also shown there that the models FLP(1, α) and FLP(2, β) are equivalent in the following sense: if all entries of the vector β are identical and equal to α, then the optimal solutions to FLP(1, α) and FLP(2, β) are the same and the optimal values differ only by a factor of $(1 - \alpha)$. The same arguments apply to the model developed in the current paper as well – introducing facility set-up costs in FLP(1, α), we obtain a model whose optimal solution can be found by solving the (NLP1) above. Due to the equivalence of the two models, we limit all subsequent discussion in the current paper to the "single-exposure" model only. However, the results apply immediately to the "multiple-exposure" version as well.

3. Construction of an equivalent linear integer program

The shortcomings of formulation (NLP1) from the computational point of view are quite apparent. In Berman et al. [4, 5], we developed a technique for obtaining equivalent Linear Integer Programming formulations for similar models. This technique is based on embedding the model in an Average or Total Reward Markov Decision Process with certain side constraints and then applying the theory of frequencyconstrained MDPs. This approach could also be applied to the model considered in the current paper, but would be quite laborious and technical. Instead, we provide below a much shorter and straightforward method resulting in the same linear IP formulation. It should be noted that since our model is a generalization of the previously considered models, the approach below could be applied to the model in Berman et al. [4,5] as well (however, we would certainly not be able to obtain the new approach without first obtaining the form of the IP and the insights from the original MDP-based approach).

Consider the following mixed integer program, which we refer to as MIP1:

(MIP1) maximize $$r\sum_{i\in N}(1-\beta_i)x_{i2} - \sum_{i\in N} s_i u_i \tag{4}$$

subject to
$$x_{i1} + x_{i2} - \sum_{j\in N} c_{ji}x_{j1} - \sum_{j\in N}\beta_j c_{ji} x_{j2} = b_i, \qquad i=1,\dots,n, \tag{5}$$
$$\sum_{i\in N} u_i \le m, \tag{6}$$
$$x_{i1} \le (1-u_i)M_{i1}, \qquad i=1,\dots,n, \tag{7}$$
$$x_{i2} \le M_{i2}u_i, \qquad i=1,\dots,n, \tag{8}$$
$$x_{i1}, x_{i2} \ge 0, u_i \in \{0,1\}, \qquad i=1,\dots,n, \tag{9}$$

where for each $i\in N$, M_{i1} is an optimal value of the following accompanying linear program, denoted as LP1:

(LP1)

$M_{i1} =$ maximize $$x_{i1} \tag{10}$$

subject to
$$x_{j1} + x_{j2} - \sum_{k\in N} c_{kj}x_{k1} - \sum_{k\in N}\beta_k c_{kj}x_{k2} = b_j, \quad j=1,\dots,n, \tag{11}$$
$$x_{j1}, x_{j2} \ge 0, \qquad j=1,\dots,n, \tag{12}$$

and M_{i2} is an optimal value of the following linear program LP2:

(LP2)

$M_{i2} =$ maximize $$x_{i2} \tag{13}$$

subject to
$$x_{j1} + x_{j2} - \sum_{k\in N} c_{kj}x_{k1} - \sum_{k\in N}\beta_k c_{kj}x_{k2} = b_j, \quad j=1,\dots,n, \tag{14}$$
$$x_{j1}, x_{j2} \ge 0, \qquad j=1,\dots,n. \tag{15}$$

Let $\boldsymbol{x}_1 = [x_{11},\dots,x_{n1}]^T$ and $\boldsymbol{x}_2 = [x_{12},\dots,x_{n2}]^T$. The equivalence between NLP1 and MIP1 is established below.

THEOREM 1

(a) If (u^*, x_1^*, x_2^*) is an optimal solution to MIP1, then u^* is an optimal solution to NLP1.

(b) If u^* is an optimal solution to NLP1, then there exist vectors x_1^*, x_2^* such that (u^*, x_1^*, x_2^*) is an optimal solution to MIP1.

(c) The optimal values of MIP1 and NLP1 are identical.

Proof

The key observation is that every feasible location vector u (i.e., a binary vector satisfying (6)) defines a *unique* feasible solution $(u, x_1(u), x_2(u))$ of MIP1. This is established as follows. Note that for any feasible location vector u, constraints (7) and (8) imply

$$x_{i1} = 0, \quad i \in F,$$
$$x_{i2} = 0, \quad i \in N - F,$$

where $F = \{i \mid u_i = 1\}$ is the set of facility nodes. Having identified n zero elements in x_1 and x_2, let us use a vector z to represent the remaining n variables, as the following:

$$z_i(u) = \begin{cases} x_{i1} & \text{if } u_i = 0, \\ x_{i2} & \text{if } u_i = 1, \end{cases} \quad i \in N.$$

The constraints (5) now represent a linear system of n equations with n unknowns, that can be written as:

$$z^T[I - C(u)] = b^T,$$

where the matrix $C(u)$ was defined in the previous section. As discussed earlier, this matrix is substochastic, and therefore the inverse $[I - C(u)]^{-1}$ is well-defined. It follows that $z^T = b^T[I - C(u)]^{-1}$ is uniquely defined. Moreover, from the definition of M_{i1}, M_{i2}, it follows that constraints (7) and (8) are satisfied by the entries of z as well. This establishes that for every feasible location vector u, there exists only one feasible solution $(u, x_1(u), x_2(u))$ to MIP1.

Using $z(u)$ to represent the non-zero entries of $x_1(u)$ and $x_2(u)$, we obtain:

$$\sum_{i \in N}(1 - \beta_i)x_{i2}(u) = \sum_{i \in N}(1 - \beta_i)u_i z_i$$
$$= b^T[I - C(u)^{-1}[(1 - \beta_1)u_1, \ldots, (1 - \beta_n)u_n]^T$$
$$= b^T[I - C(u)]^{-1}c_{n+1}(u),$$

with c_{n+1} defined in the previous section. This shows the equivalence of the objective functions of MIP1 and NLP1.

Parts (a), (b) and (c) of the theorem now follow immediately. □

Remark 3

The argument in the proof above indicates that not only are the optimal locations for NLP1 and for MIP1 identical, but also that the components $\boldsymbol{x}_1(\boldsymbol{u})$ and $\boldsymbol{x}_2(\boldsymbol{u})$ of any feasible solution of MIP1 have clear interpretations from the theory of Markov chains. In fact, $x_{i2}(\boldsymbol{u})$ represents the probability that a customer will be intercepted at a node i given the location vector $\boldsymbol{u}$ (thus $x_{i2}(\boldsymbol{u}) = 0$ if i is not a facility node), while $x_{i1}(\boldsymbol{u})$ is the expected number of visits to node i prior to interception by some facility or reaching the destination node 0 (with $x_{i1}(\boldsymbol{u}) = 0$ if i is a facility node). Indeed, it is not difficult to show that

$$x_{i1}(\boldsymbol{u}) = (\boldsymbol{b}^T[I - C(\boldsymbol{u})]^{-1})_i, \quad i \in N - F.$$

Given the above interpretations of variables, it becomes apparent that M_{i1} is the maximum expected number of visits to node i (under any location vector $\boldsymbol{u}$), and M_{i2} is the maximum probability of interception at node i. Indeed, the values of M_{i1}, M_{i2} can be computed directly (rather than by solving the linear programs LP1 and LP2) as follows. Since every facility diverts some portion of the flow to node $n + 1$, the number of expected visits to any given node $i \in N$ is maximized when no facilities are placed on the network. In that case $\boldsymbol{u} = \boldsymbol{0}$ and $C(\boldsymbol{u}) = C$ (where C is the original transition matrix corresponding to nodes $1,\dots,n$). Thus, $M_{i1} = (\boldsymbol{b}^T[I - C]^{-1})_i$. Similarly, the probability of interception at a node $i \in N$ is maximized when the facility at i is the only one on the network (since any other facility might cannibalize some of the flow). Then $\boldsymbol{u}$ is defined by $u_i = 1$, $u_j = 0$ for all $j \neq i$, and $M_{i2} = (\boldsymbol{b}^T[I - C(\boldsymbol{u})]^{-1})_i$ for $i \in N$. These observations are summarized in the following corollary:

COROLLARY 1

(a) For every $i \in N$, the optimal value M_{i1} of LP1 is given by

$$M_{i1} = (\boldsymbol{b}^T[I - C)]^{-1})_i.$$

(b) For $i \in N$, let

$$B_i = diag[1,\dots,1,\beta_i,1,\dots,1] \in R^{n\times n}.$$

Then the optimal value M_{i2} of LP2 is given by

$$M_{i2} = (\boldsymbol{b}^T[I - (B_iC)]^{-1})_i.$$

The new formulation MIP1 allows us to solve medium-sized Flow-Intercepting Facility Location Problems using commonly available integer programming software. For example, we have been able to obtain optimal solutions to 100-node (randomly generated) problems with $m = 6$ using the CPLEX software on a SUN Workstation. More details of the computational experiments are presented in section 5 below. In the next section, we analyze an approximate algorithm – the Greedy Heuristic – for our problem, and show this method to be quite an efficient procedure.

4. A greedy heuristic and worst-case analysis

In this section, we first present a simple greedy-type heuristic procedure for the Flow-Interception Problem. The procedure turns out to be highly efficient, both in terms of computational effort and accuracy, when compared to the exact solution (the results of the computational experimentation are presented in section 5 below).

For any $A \subseteq N$, let $\boldsymbol{u}^A$ be a location vector such that $u_i^A = 1$ if $i \in A$ and $u_i^A = 0$ otherwise. Let $f(A)$ be the objective function value of NLP1 corresponding to the location vector $\boldsymbol{u}^A$. The heuristic procedure is described below:

THE GREEDY-TYPE HEURISTIC ALGORITHM

Step 0: Set $k = 0$, $A_0 = \emptyset$.

Step 1: IF $k = m$, GO TO step 4.

Step 2: Set $i_{k+1} = \arg\max_{j \in N - A_k} f(A_k \cup \{j\})$. Set $A_{k+1} = A_k \cup \{i_{k+1}\}$.

Step 3: IF $f(A_k) \geq f(A_{k+1})$, GO TO step 4;
OTHERWISE, set $k = k + 1$ and GO TO step 1.

Step 4: (The Backward Search)
For $i = 1, \ldots, n$ do
IF $i \in A_k$ and $f(A_k) < f(A_k - \{i\})$ THEN
Set $A_{k-1} = A_k - \{i\}$, $k = k - 1$.

Step 5: Output the location vector $\boldsymbol{u}^{A_k}$, and the objective function value $f(A_k)$. The number of located facilities is k.

This procedure, in step 2, attempts to expand the facility set by finding a (currently empty) location with the highest profit margin. If no location with positive profit margin is available, or the limit on the available number of facilities has been reached, the procedure goes to step 4. Note that since placing new facilities might cannibalize some flow from the existing ones, as a result of the expansion of the facility set in step 2 some of the previously placed facilities might now have negative profit margins. Thus, in step 4 the facility set is checked, and the facilities with negative profit margins are discarded.

Note that steps 1–3 describe the greedy heuristic procedure for the Flow Interception Problem.

Remark 4

The "cleanup" in step 4 could be performed after every iteration of step 2, instead of just as a final step. This would result in improved accuracy, at the expense of longer running times: while this new procedure would still be finite (since the value of the objective function is required to (strictly) increase whenever the facility set is changed), the number of iterations could potentially be quite large – it is easy to construct examples where the same node enters and leaves the facility set more than once. We have chosen to do the "cleanup" step only once, as described in the Greedy-Type Heuristic Algorithm, since the accuracy levels achieved by this procedure appear to be quite high already (see section 5 for details).

We would like to derive a worst-case bound for our greedy-type algorithm by employing the submodularity-type arguments based on the results in Nemhauser and Wolsey [12] and Nemhauser et al. [13]. This technique requires the objective function of the underlying optimization problem ($f(A)$ in our case) to be monotone and submodular, i.e.,

$$f(\boldsymbol{u}^A) \geq f(\boldsymbol{u}^B) \text{ if } B \subseteq A \text{ (monotony)} \tag{16}$$

and

$$f(\boldsymbol{u}^A) + f(\boldsymbol{u}^B) - f(\boldsymbol{u}^{A\cup B}) - f(\boldsymbol{u}^{AB}) \geq 0, \text{ if } A, B \subseteq N \text{ (submodularity)} \tag{17}$$

(where $AB \equiv A \cap B$).

It will be shown below that the submodularity property holds for $f(A)$. However, in general, the monotony does not – a strict enlargement of a facility set might result in very small flows at the new facilities, leading to negative profit margins and a reduction in the objective function value (this is the reason for step 4 above). This motivates the following (rather strong) assumption:

ASSUMPTION 2

In any facility set A, every facility $i \in A$ generates a non-negative profit.

Note that under assumption 2, the heuristic algorithm presented above is exactly equivalent to the greedy heuristic, since step 4 can be skipped.

The following proposition establishes an easily-verifiable condition which is equivalent to assumption 2.

PROPOSITION 1

Let $B = diag(\beta_1, \ldots, \beta_n)$. Then assumption 2 holds if and only if

$$[\boldsymbol{b}^T(I - BC)^{-1}]_i \geq s_i / r \quad \text{for all } i \in N.$$

Proof

Note that the intercepted flow at any facility is minimized if there is a facility at every other node in the network. In this case the transition matrix of the induced Markov chain is equal to BC and the profit generated by the facility at node i is equal to

$$r[b^T(I - BC)^{-1}]_i - s_i.$$

This establishes the "if" direction.

For the "only if" direction, note that if the expression above is negative for some i, then the facility at i generates negative profit in the facility set N. □

Note that a simple (but stronger) condition that implies assumption 2 is that $s_i = 0$ for all $i \in N$. This condition is satisfied, in particular, by all models considered in Berman et al. [4,5].

We are now ready to establish a worst-case bound for the greedy-type algorithm (under assumption 2).

THEOREM 2

Let f_{greedy} be the objective function value returned by the Greedy-Type Heuristic Algorithm, and let f_{opt} be the optimal objective function value. Suppose assumption 2 holds. Then the worst-case bound on the relative error of the heuristic is given by

$$\frac{f_{greedy}}{f_{opt}} \geq 1 - \left(1 - \frac{1}{m}\right)^m \geq 1 - e^{-1}. \tag{18}$$

Proof

By Nemhauser and Wolsey [12], it suffices to show that $f(A)$ satisfies the monotony and submodularity properties (16) and (17), respectively. Under assumption 2, the monotony holds automatically.

Before establishing submodularity, let us introduce some notation. The set N can be partitioned as $N = AB \cup (A - AB) \cup (B - AB) \cup (N - (A \cup B))$, where the subsets are all disjoint. Without loss of generality, assume that $AB = \{1,\ldots, |AB|\}$, $A - AB = \{|AB| + 1,\ldots,|A|\}$, $B - AB = \{|A| + 1,\ldots, |A \cup B|\}$, and $N - (A \cup B) = \{|A \cup B| + 1,\ldots, |N|\}$.

$$C_A := C(u^A),\ C_B := C(u^B),\ C_{A\cup B} := C(u^{A\cup B}),\ C_{AB} := C(u^{AB}),$$

$$D_1 = diag[\beta_1,\ldots,\beta_{|AB|}] \in R^{|AB|\times|AB|},$$

$$D_2 = diag[\beta_{|AB|+1},\ldots,\beta_{|A|}] \in R^{|A-AB|\times|A-AB|},$$

$$D_3 = diag[\beta_{|A|+1},\ldots,\beta_{|A\cup B|}] \in R^{|B-AB|\times|B-AB|},$$

$$D_4 = diag[\beta_{|A\cup B|+1},\ldots,\beta_{|N|}] \in R^{|N-(A\cup B)|\times|N-(A\cup B)|}.$$

Further, let

$$D_{AB} = diag[D_1, I_2, I_3, I_4] \in R^{|N|\times|N|},$$

$$D_A = diag[D_1, D_2, I_3, I_4] \in R^{|N|\times|N|},$$

$$D_B = diag[D_1, I_2, D_3, I_4] \in R^{|N| \times |N|},$$

$$D_{A \cup B} = diag[D_1, D_2, D_3, I_4] \in R^{|N| \times |N|},$$

where I_2, I_3 and I_4 stand for identity matrices of dimensions $|A - AB|$, $|B - AB|$ and $|N - (A \cup B)|$, respectively. Let $C = (c_{ij}) \in R^{|N| \times |N|}$. The definitions lead to

$$C_A = D_A C,\ C_B = D_B C,\ C_{AB} = D_{AB} C,\ C_{A \cup B} = D_{A \cup B} C.$$

We further define a vector $\boldsymbol{e} \in R^{|N|}$ by $e_i = 1$ for all $i \in N$. Clearly,

$$\boldsymbol{c}_{n+1}(\boldsymbol{u}^A) = [I - D_A]\boldsymbol{e},\ \boldsymbol{c}_{n+1}(\boldsymbol{u}^B) = [I - D_B]\boldsymbol{e},$$

$$\boldsymbol{c}_{n+1}(\boldsymbol{u}^{AB}) = [I - D_{AB}]\boldsymbol{e},\ \boldsymbol{c}_{n+1}(\boldsymbol{u}^{A \cup B}) = [I - D_{A \cup B}]\boldsymbol{e}.$$

To prove submodularity (17), we write

$$f(A) + f(B) - f(A \cup B) - f(AB) = \boldsymbol{b}^T \Delta,$$

where

$$\Delta := \sum_{k=0}^{\infty} [(D_A C)^k (I - D_A) + (D_B C)^k (I - D_B) - (D_{A \cup B} C)^k (1 - D_{A \cup B}) - (D_{AB} C)^k (I - D_{AB})]\boldsymbol{e}.$$

The submodularity would be established if we can show that $\Delta \geq 0$. Let

$$\Delta_k := [(D_A C)^k (I - D_A) + (D_B C)^k (I - D_B) - (D_{A \cup B} C)^k (I - D_{A \cup B}) - (D_{AB} C)^k (I - D_{AB})]\boldsymbol{e}.$$

Obviously, $\Delta_0 = [(I - D_A) + (I - D_B) - (I - D_{A \cup B}) - (I - D_{AB})]\boldsymbol{e} = 0$. This leads to

$$\Delta = \sum_{k=0}^{\infty} \Delta_k = \sum_{k=0}^{\infty} \Delta_{k+1}. \tag{19}$$

Now

$$\begin{aligned}\Delta_{k+1} &= [(D_A C)^{k+1} (I - D_A) + (D_B C)^{k+1} (I - D_B) \\ &\quad - (D_{A \cup B} C)^{k+1} (I - D_{A \cup B}) - (D_{AB} C)^{k+1} (I - D_{AB})]\boldsymbol{e} \\ &= D_{A \cup B} C [(D_A C)^k (I - D_A) + (D_B C)^k (I - D_B) \\ &\quad - (D_{A \cup B} C)^k (I - D_{A \cup B}) - (D_{AB} C)^k (I - D_{AB})]\boldsymbol{e} \\ &\quad + [(D_A - D_{A \cup B}) C (D_A C)^k (I - D_A) + (D_B - D_{A \cup B}) C (D_B C)^k (I - D_B) \\ &\quad - (D_{AB} - D_{A \cup B}) C (D_{AB} C)^k (I - D_{AB})]\boldsymbol{e}\end{aligned}$$

$$= D_{A\cup B}C\Delta_k$$
$$+ [(D_A - D_{A\cup B})C(D_AC)^k(I - D_A) + (D_B - D_{A\cup B})C(D_BC)^k(I - D_B)$$
$$- (D_{AB} - D_{A\cup B})C(D_{AB}C)^k(I - D_{AB})]\boldsymbol{e}.$$

Using the relation $D_{AB} - D_{A\cup B} = (D_A - D_{A\cup B}) + (D_B - D_{A\cup B})$, we are left with

$$\Delta_{k+1} = D_{A\cup B}C\Delta_k + (D_A - D_{A\cup B})C[(D_AC)^k(I - D_A) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e}$$
$$+ (D_B - D_{A\cup B})C[(D_BC)^k(I - D_B) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e}.$$

Taking summation over $k = 0, 1, 2, \ldots$, we obtain

$$\Delta = D_{A\cup B}C\Delta$$
$$+ (D_A - D_{A\cup B})C\sum_{k=0}^{\infty}[(D_AC)^k(I - D_A) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e}$$
$$+ (D_B - D_{A\cup B})C\sum_{k=0}^{\infty}[(D_BC)^k(I - D_B) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e}. \quad (20)$$

We need to establish that

$$\sum_{k=0}^{\infty}[(D_AC)^k(I - D_A) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e} \geq 0, \quad (21)$$

$$\sum_{k=0}^{\infty}[(D_BC)^k(I - D_B) - (D_{AB}C)^k(I - D_{AB})]\boldsymbol{e} \geq 0. \quad (22)$$

Note that if (21) and (22) hold, then the proof can be completed as follows: since $D_A - D_{A\cup B} \geq 0$ and $D_B - D_{A\cup B} \geq 0$, it follows from (20) that

$$(I - D_{A\cup B}C)\Delta \geq 0.$$

Because $D_{A\cup B}C = C(\boldsymbol{u}^{A\cup B})$ is a substochastic matrix,

$$(I - D_{A\cup B}C)^{-1} \geq 0,$$

which implies $\Delta \geq 0$, completing the proof. It only remains to establish the validity of (21) and (22).

To establish (21) we proceed as follows

$$\sum_{k=0}^{\infty}(D_AC)^k(I - D_A) - (D_{AB}C)^k(I - D_{AB})$$
$$= (I - D_AC)^{-1}(I - D_A) - (I - D_{AB}C)^{-1}(I - D_{AB})$$

$$= (I - D_A C)^{-1}(I - D_A C + D_A C - D_A)$$
$$- (I - D_{AB}C)^{-1}(I - D_{AB}C + D_{AB}C - D_{AB})$$
$$= (I - D_A C)^{-1} D_A (C - I) - (I - D_{AB}C)^{-1} D_{AB}(C - I)$$
$$= [(I - D_{AB}C)^{-1} D_{AB} - (I - D_A C)^{-1} D_A](I - C).$$

Thus, to verify (21) it suffices to verify that

$$[(I - D_{AB}C)^{-1} D_{AB} - (I - D_A C)^{-1} D_A](I - C)e \geq 0.$$

Note that since $D_{AB} \geq D_A$, we have

$$(I - D_{AB}C)^{-1} D_{AB} - (I - D_A C)^{-1} D_A = \sum_{k=0}^{\infty} [(D_{AB}C)^k D_{AB} - (D_A C)^k D_A] \geq 0,$$

and since C is a substochastic matrix, $(I - C)e \geq 0$. This establishes (21). The inequality (22) is established similarly. □

Note that the proof of submodularity above did not require assumption 2. The next result proves the tightness of the worst-case bound given in theorem 2.

THEOREM 3

For any $n \geq 3m - 1 > 2$, there exists an instance of the Flow-Intercepting Facility Location Problem for which the worst-case bound of theorem 2 is tight for the Greedy-Type Heuristic Algorithm.

Proof

We construct a network, as depicted in figure 2, for which the greedy heuristic solution could reach the worst-case bound of $1 - (1 - 1/m)^m$. We let $\beta_i = s_i = 0$ for all i (but the case where $\beta_i = \beta > 0$ could be handled similarly with minor modifications). In figure 2, the transition probabilities are indicated next to each arc. Note that nodes $m + 1$ through $3m - 1$ all have a single outgoing arc to the common sink node 0 (not shown in figure 2).

The initial distribution (b_i) is given by

$$b_i = \begin{cases} 1/m & i = 1, \ldots, m, \\ 0 & i = m + 1, \ldots, 3m - 1, \end{cases}$$

and the transition probabilities (c_{ij}'s) are defined as follows:

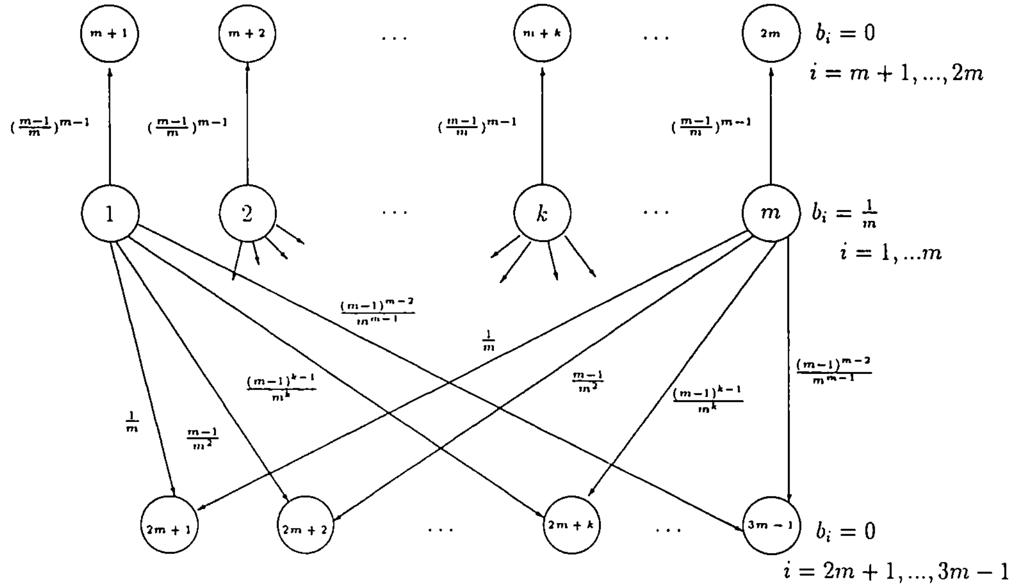

Figure 2. Network for theorem 3.

$$c_{ij} = \begin{cases} \frac{(m-1)^{k-1}}{m^k} & i = 1, \ldots, m;\ j = 2m+k,\ k = 1, \ldots, m-1, \\ \left(\frac{m-1}{m}\right)^{m-1} & i = 1, \ldots, m;\ j = m+i, \\ 1 & i = j,\ i \in \{m+1, m+2, \ldots, 3m-1\}, \\ 0 & \text{otherwise.} \end{cases}$$

With this network, the optimal solution for NLP1 is $u_1^* = \cdots = u_m^* = 1$ and $u_{m+1}^* = \cdots = u_{3m-1}^* = 0$, and the total customer flow intercepted optimally is $f(u^*) = 1$ (= total flow available in the network).

Note that since assumption 2 is satisfied by the network defined above, our heuristic procedure is exactly equivalent to the greedy heuristic, i.e., step 4 can be skipped and the heuristic will always locate exactly m facilities.

To apply the greedy heuristic, let us define

i_k = the location (node) chosen by the heuristic at iteration k, $k = 1, \ldots, m$,

f_k = total intercepted customer flow at iteration k,

Δf_k = marginal increase in the total intercepted flow at iteration k.

Next, we apply the greedy heuristic step by step to the network above. From the network structure, we see that locating at node $i \in \{1, \ldots, m\}$ is always at least as good as locating at node $m+j, j \in \{1, \ldots, m\}$; therefore, nodes in the set $\{m+1, \ldots, 2m\}$ can be eliminated from consideration. We will show that the heuristic could choose the facility set $\{2m+1, 2m+2, \ldots, 3m-1\} \cup i$ with i arbitrarily selected from the set $\{1, \ldots, m\}$. Note that since all set-up costs are set to 0, the objective function is linear in the total flow intercepted. Thus, we can take the total intercepted flow as the objective function when applying the heuristic.

First facility:

If located at any $i \in \{1, \ldots, m\}$, $f_1 = 1/m$;
if located at $i \in \{2m+1, \ldots, 3m-1\}$, the best choice is $i = 2m+1$ with $f_1 = 1/m$;
Therefore, the heuristic could choose $i_1 = 2m+1$ with $f_1 = 1/m$.

Second facility:

If located at any $i \in \{1, \ldots, m\}$, $\Delta f_2 = (1/m)(1 - 1/m) = (m-1)/m^2$;
if located at $i \in \{2m+2, \ldots, 3m-1\}$, the best choice is $i = 2m+2$ with $\Delta f_2 = (m-1)/m^2$;
Therefore, the heuristic could choose $i_2 = 2m+2$ with $\Delta f_2 = (m-1)/m^2$;

In an inductive argument, we assume that the first $k-1$ $(2 \le k < m-1)$ facilities are located at nodes $2m+1, \ldots, 2m+k-1$, respectively. Now we proceed to the kth facility.

kth facility:

If located at any $i \in \{1, \ldots, m\}$ (note that no facilities have been placed at nodes $1, \ldots, m$ by the inductive hypothesis),

$$\begin{aligned}\Delta f_k &= \frac{1}{m}\left(1 - \sum_{l=1}^{k-1} \frac{(m-1)^{l-1}}{m^l}\right) \\ &= \frac{1}{m}\left(1 - \frac{1}{m}\,\frac{1 - ((m-1)/m)^{k-1}}{1 - (m-1)/m}\right) = \frac{(m-1)^{k-1}}{m^k};\end{aligned}$$

if located at $i \in \{2m+k, \ldots, 3m-1)$, the best choice is $i = 2m+k$ with $\Delta f_k = (m-1)^{k-1}/m^k$. Therefore, the heuristic could choose $i_k = 2m+k$ with $\Delta f_k = (m-1)^{k-1}/m^k$.

It now becomes clear that the heuristic may choose nodes $2m+1, \ldots, 3m-1$ as the first $m-1$ locations. At this point, the last facility (the mth one) can be located anywhere among nodes $1, \ldots, m$ since any single location in this node set gives

$$\Delta f_m = \frac{1}{m}\left(\frac{m-1}{m}\right)^{m-1}.$$

The total flow intercepted by the heuristic in the above fashion is thus

$$f_m = f_1 + \sum_{i=2}^{m} \Delta f_i = 1 - \left(\frac{m-1}{m}\right)^m,$$

and since the total flow intercepted by the optimal solution is 1, the relative error is equal to the worst-case bound of theorem 2. □

Remark

The worst-case bound holds for the greedy-type heuristic when the set-up cost is sufficiently low (assumption 2, which requires the marginal profit at any facility location to be non-negative for any $\boldsymbol{u}$). It is of interest to know what happens to the performance of the heuristic if assumption 2 is violated. It would be desirable if the worst-case result can be extended to this general case, similar to the work of Cornuéjols et al. [7], where the result is first proved with zero fixed cost and then extended to the general case. Unfortunately, assumption 2 seems to be necessary for the worst-case bound to hold in our case, as illustrated in the following example.

EXAMPLE

Consider the 7-node network shown in figure 3 (similar examples for general numbers of facilities can be easily constructed). The c_{ij}, b_i, s_i information is provided in the graph. We assume that there are $m = 3$ facilities and $r = 10$.

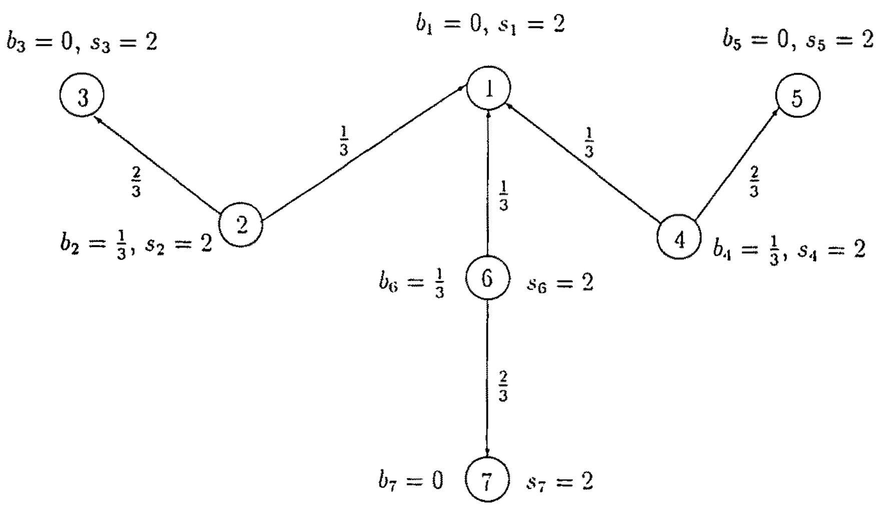

Figure 3. Example: Woest-case violated when assumption 2 does not hold.

Let $f(k_1, k_2, \ldots)$ represent the objective function value when facilities are located at nodes $k_1, k_2, \ldots$. Applying the heuristic, we have the following steps:

Step 1. $f(1) = f(2) = f(4) = f(6) = 4/3$, $f(3) = f(5) = f(7) = 2/9$. Suppose the first location chosen is node 1.

Step 2. $f(1, 2) = f(1, 4) = f(1, 6) = f(1, 3) = f(1, 5) = f(1, 7) = 14/9$. Suppose the second location chosen is node 2.

Step 3. $f(1, 2, 4) = f(1, 2, 6) = 16/9$, and all other choices give an objective function value less than 16/9. Suppose the third location chosen is node 4.

Step 4. Since all $m = 3$ facilities are located, we perform Backward Search in step 4. It is found that the facilities at nodes 2 and 4 both have a positive profit margin, whereas the facility at node 1 has a negative margin of $-8/9$. Thus, node 1 is deleted as a facility location. We end up with 2 facility locations (nodes 2 and 4) and an objective function value of 8/3.

An optimal solution to the problem is locating three facilities at nodes 2, 4 and 6 and achieving an objective function value of 4. Since the heuristic locates only $2 < m = 3$ facilities, it does not even properly address the original optimization problem, let alone a worst-case bound. The reason, obviously, is due to step 4 of the algorithm – any facility location with a non-positive profit margin is deleted and no effort is made to relocate the facility elsewhere by the algorithm.

To solve this problem, as noted in remark 3, an alternative greedy-type algorithm might be used which does the cleanup (step 4) after each expansion of the facility set (and, which achieves the optimal solution for this example). This, however, as noted earlier, may result in significantly lengthening the running time. From our limited computational experience (section 5), it seems that the present version of the greedy heuristic already works quite well.

5. Some computational results

In this section, we present computational results obtained with several testing examples. These tests are designed to show (1) the performance of the greedy heuristic, and (2) how model parameters may affect the computational efforts needed for finding an exact solution. The software package for solving the MIP is CPLEX Mixed Integer Solver. All the examples were run on a SUN SPARC II workstation. All codes are written in C.

In the examples, the data c_{ij} and b_i, $(i, j \in N)$ are all generated randomly, first using a uniform distribution in the range [0, 1], and then normalizing the data such that $\sum_{j=0}^{n} c_{ij} = 1\ \forall i$ and $\sum_{i \in N} b_i = 1$. For most of the examples (tables 1, 2, 3, 5 for s_i and tables 1, 3, 4, 5 for β_i), $\{s_i, i \in N\}$ are generated randomly using a uniform distribution in the range [0, 20], and $\{\beta_i, i \in N\}$ are generated randomly using a uniform distribution in the range [0, 1].

The results contained in tables 1–5 are based on averages of a sample size of 5 runs for each problem instance (given in a row of a corresponding table). Table 1

Table 1

Comparison of the objective function values and CPU times – greedy heuristic and CPLEX, random s_i, β_i, $r = 1000$.

		T_{CPU} (sec)		
n, m	J_{GH}/J_{opt}	CPLEX	Greedy	n_{BB}
20, 4	0.998	1.2	0.9	87
40, 4	0.999	22.8	13.7	671
60, 4	0.999	94.7	73	1364
80, 4	0.99	160	230	1052
100, 4	1	1006	559.3	7993
100, 7	NA	NA	963	
100, 10	NA	NA	1370	

Table 2

Varying $(1/n)\sum \beta_i$, $r = 1000$, $(n, m) = (40, 4)$, random s_i.

		T_{CPU} (sec)		
$(1/n)\sum_i \beta_i$	J_{GH}/J_{opt}	CPLEX	Greedy	n_{BB}
0	1	4.7	13.7	8
0.2	0.943	7.5	13.7	122
0.4	1	21.1	13.7	785
0.6	1	16.7	13.7	574
0.8	1	10.8	13.7	332

Table 3

Varying r, random s_i and β_i, $(n, m) = (40, 4)$.

		T_{CPU} (sec)		
r	J_{GH}/J_{opt}	CPLEX	Greedy	n_{BB}
50	1	7	13.7	135
100	1	10.1	13.7	278
200	0.997	13	13.7	413
400	1	16	13.7	563
600	1	16.5	13.7	591
800	1	16.8	13.7	576
1000	0.999	18.4	13.7	671

Table 4

Comparison of the objective function values and CPU times – greedy heuristic and CPLEX, $s_i = 0$, $r = 1000$.

n, m	J_{GH}/J_{opt}	T_{CPU} (sec) CPLEX	Greedy	n_{BB}
20, 4	0.999	1.4	0.9	105
40, 4	1	22.9	13.7	723
60, 4	0.999	85.8	73	1211
80, 4	0.991	147	230	730
100, 4	0.999	1067	559.3	6292
100, 7	NA	NA	974	
100, 10	NA	NA	1369	

Table 5

Comparison of the objective function values and CPU times random β_i and s_i, $r = 200$ – greedy heuristic and CPLEX.

n, m	J_{GH}/J_{opt}	T_{CPU} (sec) CPLEX	Greedy	n_{BB}
20, 4	1	0.8	0.9	61
40, 4	0.997	13.2	13.4	413
60, 4	0.994	56.8	76	877
80, 4	0.99	104.2	245	567
100, 4	0.999	344.5	569.3	1441
20, 6	0.999	2.8	1.3	303
40, 6	1	136.9	21.4	5999
60, 6	1	1390	115.2	30059
80, 6	1	5044	343.9	57828
100, 6		4000	855	

compares the objective function values and CPU times of the greedy heuristic and the exact (CPLEX) MIP for various problems sizes represented by different values of n and m. The parameter n_{BB} is the number of partial solutions (LPs) solved during the branch-and-bound. Table 2 shows how the obscureness indices β_i (represented by $(1/n)\sum_{i=1}\beta_i$) affect the performance of the greedy heuristic and the MIP (when the average of the indicies is small, the facilities are quite efficient at capturing the bypassing flow, and vice versa). Table 3 compares the performance of the greedy heuristic and the MIP when the revenue coefficient r is varied. Recall that the value of r represents the maximum value that could be earned by capturing all available flow on the network. Thus, since $s_i = 10$ on the average, and $m = 4$, for $r = 50$, the set-up costs play a major role, while for $r = 1000$ the set-up costs are mostly irrelevant. Tables 4 and 5 are similar to table 1 except for the setup costs s_i which are all equal to 0 in table 4 and the revenue coefficient r which is much smaller in table 5 than in table 1. Thus, table 4 attempts to analyze the performance of the heuristic for the "no set-up cost case" (e.g., as in the models considered in our previous work), while table 5 examines the performance when the set-up costs do play a major role.

Based on the results contained in tables 1–5 the following conclusions are drawn:

(1) The greedy heuristic achieves optimal or close to optimal solutions for all the problems tested, regardless of the values of β_i, s_i, n, m and r.

(2) When varying $(1/n)\sum\beta_i$, the problem is more difficult to solve for intermediate values of β_i than for extreme values of β_i. Recalling that $(1/n)\sum\beta_i = 0$ is equivalent to the original problem presented in Berman et al. [4], where 100% of customers are intercepted by a facility node, and that large values of β_i correspond to the case when very few customers are intercepted, the conclusion is quite intuitive.

(3) As intuitively expected, when r is increasing the problem becomes more difficult to solve.

(4) The heuristic is much more efficient than the MIP for large values of n and m.

References

[1] O. Berman, The maximizing market size with congestion, Socio-Economic Planning Sciences 29(1995)39–46.

[2] O. Berman, D. Bertsemas and R.C. Larson, Locating discretionary service facilities II: Maximizing market size, minimizing inconvenience, accepted by Operations Research.

[3] O. Berman, R.C. Larson and N. Fouska, Optimal location of discretionary service facilities, Transportation Science 26(23) (1992) 201–211.

[4] O. Berman, D. Krass and C.W. Xu, Locating discretionary service facilities based on probabilistic customer flows, Transportation Science 29(1995)276–290.

[5] O. Berman, D. Krass and C.W. Xu, Generalized discretionary service facility location models with probabilistic customer flows, Working Paper (1994), revised for Stochastic Models.

[6] E. Çinlar, *Introduction to Stochastic Processes* (Prentice–Hall, NJ (1974).

[7] G. Cornuéjols, M.L. Fisher and G.L. Nemhauser, Locations of bank accounts to optimize float: An analytic study of exact and approximate algorithms, Management Science 23(1977)789–809.

[8] R.B. Dial, A probabilistic multi path traffic asssignment model which obviates path enumeration, Transportation Research 5(1971)83–111.

[9] J. Hodgson, A flow-capturing location-allocation model, Geographic Analysis 22(3) (1990) 270–279.

[10] L.C.M. Kallenberg, *Linear Programming and Finite Markovian Control Problems*, Math. Centre Tracks 148(1983).

[11] R.S. Lewis and T.W. Leznick, A report on Boeing fleet location and information reporting system, The Boeing Company, Wichita, Kansas, presented at the *10th Annual Carnaham Crime Countermeasures Conference*, University of Kentucky, Lexington, KY (1976).
[12] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).
[13] G.L. Nemhauser, L.A. Wolsey and M.L. Fisher, An analysis of the approximations for maximizing submodular set functions, Mathematical Programming 14(1978)265–294.
[14] Y. Sheffi, *Urban Transportation Networks* (Prentice Hall, 1985).