Les Cahiers du GERAD ISSN: 0711–2440

The Maximum Return-On-Investment Plant Location Problem With Market Share

J. Brimberg, P. Hansen,
G. Laporte, N. Mladenović,
D. Urošević

G–2005–25

March 2005

The Maximum Return-On-Investment Plant Location Problem With Market Share

Jack Brimberg

GERAD and Royal Military College of Canada
Kingston (Ontario) Canada

Pierre Hansen

GERAD and HEC Montréal
Montréal (Québec) Canada

Gilbert Laporte

GERAD, CRT and HEC Montéal
Montréal (Québec) Canada

Nenad Mladenović

GERAD and School of Mathematics, University of Birmingham
Edgbaston, United Kingdom

Dragan Urošević

Mathematical Institute, SANU
Belgrade, Serbia

March 2005

Les Cahiers du GERAD

G–2005–25

Abstract

This paper examines the plant location problem under the objective of maximizing return-on-investment. However, in place of the standard assumption that all demands must be satisfied, we impose a minimum acceptable level on market share. The model presented takes the form of a linear fractional mixed integer program. Based on properties of the model, a local search procedure is developed to solve the problem heuristically. Variable neighborhood search and tabu search heuristics are also developed and tested. Thus, a useful extension of the simple plant location problem is examined, and heuristics are developed for the first time to solve realistic instances of this problem.

Key Words: Plant location, investment, market share, fractional programming, heuristics.

Résumé

On étudie le problème de localisation d'usines avec l'objectif de la maximisation du retour-sur-investissement. Toutefois, au lieu de l'hypothèse usuelle que toutes les demandes doivent être satisfaites, nous imposons un niveau minimum acceptable sur la part de marché. Le modèle proposé prend la forme d'un programme linéaire fractionnaire en variables mixtes. Sur la base de propriétés de ce modèle, on développe une procédure de recherche locale pour résoudre le problème de manière heuristique. Des heuristiques de recherche à voisinage variable et de recherche avec tabous sont également développées et testées. De la sorte, on examine une extension utile du problème simple de localisation des usines et on développe pour la première fois des heuristiques permettant de résoudre des instances réalistes de ce problème.

Mots clés : Localisation d'usines, retour-sur-investissement, part de marché, programmation linéaire fractionnaire, heuristiques.

1 Introduction

The simple plant location problem requires choosing facility sites on a network to open in order to serve a given set of customers with known demands at minimum total cost. Thus, the objective is to find the best balance (or tradeoff) between the fixed costs associated with opening facilities and the variable costs associated with supplying and transporting the product or service to the market. This problem, beginning with the original formulation by Balinski (1965) (also attributed to Stollsteimer, 1963, and Kuehn and Hamburger, 1963), has received considerable attention in the literature on location theory, and several exact and approximate methods have been proposed to solve it. See, for example, the surveys in Krarup and Pruzan (1983), Labbé et al. (1995), Labbé and Louveaux (1997), and relevant chapters in Mirchandani and Francis (1990), Francis et al. (1992) and Daskin (1995).

The inherent assumptions of the basic model may severely restrict its usefulness in practice. ReVelle and Laporte (1996) suggest that minimizing total cost should be replaced by maximization of the return-on-investment (ROI) as the real objective in an industrial application. This latter objective is investigated in Myung et al. (1997) and Brimberg and ReVelle (2000). In both papers, the solution that maximizes ROI is obtained by solving a related bicriterion model. Myung et al. use profitability and ROI as the two criteria to be maximized. However, the problem is converted to a parametric single objective uncapacitated facility location problem having the same form as the basic model, and which is shown to produce the same set of efficient points as by the weighting method. Brimberg and ReVelle use profitability and the investment cost as their two criteria (other models are also suggested), and apply the weighting method directly. In both cases, the maximum ROI solution is one of the efficient points obtained in the solution process.

The basic model makes the assumption that all demands must be satisfied. However, in a real situation, some customers may be unprofitable or only marginally profitable (e.g., these customers have remote locations), and hence, they should not be included in the picture (see, e.g., Brimberg and ReVelle, 1998, for further discussion on this subject). The solution that maximizes profitability or ROI, therefore, generally captures less than 100% of market share. Brimberg and ReVelle (2000) suggest that the requirement to satisfy all demands be relaxed; however, without a constraint on market share, the solution will not guarantee an adequate amount of profit. In fact, there always exists an optimal solution in which only the facility offering the highest return on investment is opened (ReVelle and Laporte, 1996). Myung et al. (1997), in the context of their bicriterion model, also remove the requirement to satisfy all demands. An additional constraint imposes a minimum acceptable level of profit.

The aim of this paper is to present a formal model that maximizes return-on-investment subject to an acceptable lower bound on percent of market share in line with the recommendation in ReVelle and Laporte (1996). By parameterizing the percent of market share, the model further allows the decision-maker to examine the tradeoff on ROI over the full range of market share. We also propose some powerful heuristics to solve the model directly, rather than finding the efficient points of a related bicriterion model, as suggested in earlier papers. Whereas these earlier papers consider exact solution methods and apply them only to small instances of their linearized models, our heuristics allow much larger problem sizes to be investigated with reasonable requirements on computing time and memory.

The remainder of the paper is organized as follows. In the next section, we present the maximum ROI model with market share, and derive some results that are useful for solving it directly. A local search procedure is proposed in Section 3 for finding local optima of the model. The local search is subsequently used in Variable Neighborhood and Tabu Search heuristics developed in Section 4. Computational experience is discussed in Section 5, followed by some conclusions and directions for future research.

2 The model

We introduce a similar notation as given in ReVelle and Laporte (1996). Consider the following input data:

$i, I = \{1, \ldots, m\}$, denote the index and set of eligible plant sites;
$j, J = \{1, \ldots, n\}$, the index and set of demand points or customers;
f_i = fixed cost for opening a new facility at site i;
e_i = fixed expansion cost per unit produced at site i;
g_i = variable cost per unit produced at i (labor, materials, overhead, etc.);
c_{ij} = cost to deliver customer j's full demand from site i;
d_j = demand for the product at j;
p_j = price per unit sold to customer j;

opening costs and demands are assumed to be positive ($f_i > 0$, $\forall i$, $d_j > 0$, $\forall j$); all other parameters have nonnegative values. All costs relate to the same time horizon.

The decision variables are given by:

y_i, a zero-one variable, equal to 1 if a new facility is opened at site i, and 0 otherwise;
x_{ij} = fraction of j's demand supplied by a facility at site i.

Using this notation, the ROI plant location problem with market share may be formulated as follows:

$$\max z(X, Y) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (p_j d_j - c_{ij} - g_i d_j)\, x_{ij}}{\sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} e_i d_j x_{ij}} \tag{ROI}$$

s.t.

$$\sum_{i=1}^{m} x_{ij} \ \leq 1, \forall j \in J, \tag{1}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_j x_{ij} \ \geq \ D\alpha, \tag{2}$$

$$0 \leq x_{ij} \ \leq \ y_i \quad \forall i \in I, \ \forall j \in J, \tag{3}$$

$$y_i \in \{0, 1\}, \quad \forall i \in I, \tag{4}$$

where $D = \sum_{j \in J} \ d_j$ is the total market share, and $0 \leq \alpha \leq 1$ denotes a minimum acceptable fraction. The numerator in the objective function gives the net profit (total revenue - variable costs), normally expressed on an annual basis, while the denominator gives the total fixed investment as a net present value. The ratio of the two sums estimates the return on investment as a function of the decision variables (X, Y). Note that the constraints in (1) now imply that all demands do not have to be satisfied. However, the total demand satisfied must meet a minimum acceptable percentage of market share as specified in (2). The constraints in (3) imply that any site i must be opened ($y_i = 1$) if a positive flow emanates from that site.

Setting $\alpha = 0$ effectively removes constraint (2) from the problem. Then, if the net profit per unit,

$$\pi_{ij} = p_j - c_{ij}/d_j - g_i > 0, \tag{5}$$

for some pair (i, j), it is clear that the optimal solution to (ROI) must include some positive production. Otherwise, if $\pi_{ij} < 0$, $\forall (i, j)$, we obtain the trivial solution of keeping all production sites closed. Assuming (5) holds for some (i, j), we obtain the following known but unproven result.

Property 1 *An optimal solution of the (ROI) problem without market share is given by a single open facility site that offers the highest return on investment.*

Proof. Solve (ROI) without constraint (2) and with only site i open to obtain $z(i) = P_i/V_i$, where P_i and V_i are the corresponding profit and investment. Repeat for $i = 1, \ldots, m$, and reorder the indices so that $z(1) \geq z(2) \geq \cdots \geq z(m)$.

Next consider the problem with sites 1 and 2 open, and no others, and let $z(1, 2) = (P(1)+P(2))/(V(1)+V(2))$ denote the optimal solution, where $P(1), P(2)$, and $V(1), V(2)$ are respective profit contributions and investments. It is clear that

$$z(1) \geq \max\{\frac{P(1)}{V(1)}, \frac{P(2)}{V(2)}\}.$$

Furthermore, by elementary algebra, we know that

$$\max\{\frac{P(1)}{V(1)}, \frac{P(2)}{V(2)}\} \geq \frac{P(1)+P(2)}{V(1)+V(2)}.$$

Thus, $z(1) \geq z(1,2)$. We may use an inductive argument to show that max ROI cannot exceed $z(1)$ for any combination of open sites. □

Algorithm 1 (exact solution of ROI without market share constraint)

Step 1. *For* $i = 1, \ldots, m$ do:
(1) calculate unit profit from customer j, π_{ij}, $j = 1, \ldots, n$;
(2) arrange π_{ij} values in nonincreasing order,

$$\pi_i^{(1)} \geq \pi_i^{(2)} \geq \cdots \geq \pi_i^{(n)};$$

(3) set $P = 0$, $V = f_i$, $z(i) = 0$, $\ell(i) = 0$;
(4) *for* $\ell = 1, \ldots, n$ do:
(a) $P = P + \pi_i^{(\ell)} d_i^{(\ell)}$, $V = V + e_i d_i^{(\ell)}$, $z = P/V$;
(b) if $z \geq z(i)$, set $z(i) = z$, and $\ell(i) = \ell$;
else return last solution for site i.

Step 2. *Set* $i^* = \arg\max\{z(i), i = 1, \ldots, m\}$.
Step 3. *Retain* solution found for i^* as optimal. □

Thus, we have a simple $O(mn \log n)$ algorithm to solve (ROI) without constraint (2). Furthermore, the total demand satisfied in the obtained solution provides a lower bound on percentage of market share:

$$\alpha_{min} = \frac{\sum_{\ell=1}^{\ell(i^*)} d_{i^*}^{(\ell)}}{D}. \tag{6}$$

Now consider (ROI) with the market share constraint (2) reinstated, and a specified subset $I_0 \subseteq I$ of open facility sites; that is, $y_i = 1$, $\forall i \in I_0$, and $y_i = 0$, $\forall i \in I \setminus I_0$. Let $K = \sum_{i \in I_0} f_i$ represent the total opening cost for this solution. We may rewrite (ROI) as follows:

$$\max z_0(X) = \frac{\sum_{i \in I_0} \sum_{j=1}^{n} \pi_{ij} d_j x_{ij}}{K + \sum_{i \in I_0} \sum_{j=1}^{n} e_i d_j x_{ij}} \tag{ROI-1}$$

s.t.

$$\sum_{i \in I_0} x_{ij} \leq 1, \forall j \in J, \tag{7}$$

$$\sum_{i \in I_0} \sum_{j=1}^{n} d_j x_{ij} \ \geq \ D\alpha, \tag{8}$$

$$x_{ij} \ \geq \ 0, \quad \forall i \in I_0, \, \forall j \in J. \tag{9}$$

Next convert (ROI-1) into the following associated parametric problem:

$$\max \left\{ \sum_{i \in I_0} \sum_{j=1}^{n} \pi_{ij} d_j x_{ij} - s(K \ + \sum_{i \in I_0} \sum_{j=1}^{n} e_i d_j x_{ij}) \right\} \tag{P_1}$$

s.t. (7), (8) and (9),

where $s \in R$ is a parameter. Let us denote the optimal value of the objective function in (P_1) by $F(s)$. It is well known that solving (ROI-1) is equivalent to finding the unique root of the nonlinear equation $F(s) = 0$ (Dinkelbach, 1967).

Suppose now that we fix the value of s. The constant $-sK$ may be deleted from the objective function, leaving us with:

$$\max \left\{ \sum_{i \in I_0} \sum_{j=1}^{n} (\pi_{ij} - s e_i) d_j x_{ij} \right\} \tag{P_2}$$

s.t. (7), (8) and (9). It is readily seen that (P_2) may be solved by a *greedy* approach. That is, the unit profit coefficients

$$\xi_{ij} = \pi_{ij} - s e_i \tag{10}$$

are ranked in nonincreasing order, given by

$$\xi_{i_1 j_1} \geq \xi_{i_2 j_2} \geq \cdots \geq \xi_{i_{tn} j_{tn}} \tag{11}$$

where $t = |I_0|$. Then we assign the customer demands once (to satisfy (7)) in order of highest unit profitability until the ξ_{ij} values become negative. If the market constraint is satisfied, stop; else continue to add customer demands in order until the constraint is just satisfied. Since each unit of demand is added to the solution in order of greatest profitability, it is clear that the greedy solution must be optimal.

Property 2 *Given that the $m \times n$ coefficients ξ_{ij} are arranged in nonincreasing order, problem (P_2) is solvable in $O(mn)$ time.*

Proof. Observe that the sequence in (11) may be constructed in linear time in the number of elements $(m \times n)$ of the larger sequence. Since this step governs the complexity of the procedure for (P_2), the result immediately follows. □

Now the resolution of $F(s) = 0$ (and therefore solving (ROI-1)) is readily achieved. Noting that the objective function is convex piecewise linear, we may, for example, apply directly the Newton-Raphson method (also known as Dinkelbach's method) as follows.

Algorithm 2 (solving ROI-1)

Step 1. *Choose* an initial value, s_0 (e.g., $s_0 = 0$), and set iteration counter $\ell = 0$.

Step 2. *Solve* (P_2) for $s = s_\ell$; let $X_\ell = (x_{ij}^\ell)$ denote the obtained optimal solution. If $F(s_\ell) = 0$, stop.

Step 3. *Determine* the new value of s,

$$s_{\ell+1} = s_\ell + \frac{F(s_\ell)}{\left(K + \sum_{i \in I_0} \sum_{j=1}^{n} e_i d_j x_{ij}^\ell\right)};$$

set $\ell = \ell + 1$, and return to step 2. $\square$

It is interesting to note that parameter "s" has an economic interpretation as a discount factor on the investment. Furthermore, $F(s)$ may be viewed as an annual (or per period) net profit after discounting the investment cost. The value s_l obtained at the end of algorithm 2 ($F(s_l) = 0$) then gives the internal rate of return achievable for the specified subset I_0 of open facilities, i.e., the final value of s gives the best return on investment that may be achieved with I_0.

3 A local search procedure for (ROI)

From the preceding section, we see that once the subset of facility sites that are opened, $I_0 \subseteq I$, is given, the related problem (ROI-1) may be solved exactly and efficiently using algorithm 2. Furthermore, for an assumed value of s, the related subproblem (P_2) is solvable in $O(mn)$ time (see property 2). This insight may be readily incorporated in a heuristic procedure for finding local optima in the original problem (ROI).

The proposed local search borrows the standard *drop/add/interchange* neighborhood from the simple plant location problem (e.g., see Kuehn and Hamburger, 1963, Manne, 1964, Feldman et al., 1966, Teitz and Bart, 1968, Arya et al., 2001, Ghosh, 2003, and Hansen et al., 2003, among others). That is, given a current solution with open facilities contained in I_0, we consider all neighborhood points obtained by one of the following moves:

(i) closing an open site, $i_1 \in I_0$ ($I_0 := I_0 \setminus \{i_1\}$),
(ii) opening a closed site, $i_2 \in I \setminus I_0$ ($I_0 := I_0 \cup \{i_2\}$), or
(iii) interchanging an open site i_1 with a closed site i_2 ($I_0 := I_0 \setminus \{i_1\} \cup \{i_2\}$).

Denote the current solution as x_0, and the given neighborhood as $\mathcal{N}_{loc}(x_0)$. The solution x_0 is characterized by the set of open facility sites I_0, and the vector (X_f, s_f) obtained by solving (ROI-1). We see that there are generally $(m + t(m - t))$ points in $\mathcal{N}_{loc}(x_0)$, where again, $t = |I_0|$, with the exception being $t = 1$ (move (i) not allowed) where the number in the formula is reduced by 1.

Two variants of local search are outlined below for the established neighborhood $\mathcal{N}_{loc}$. The first solves the related problem (ROI-1) exactly at each neighborhood point:

Local search 1 (LS-1)

Step 1. *Set* $x^* := x_0$.
Step 2. *Repeat* for each point in $\mathcal{N}_{loc}(x_0)$:
solve (ROI-1) with algorithm 2 to obtain solution x.
if ROI$(x) >$ ROI (x^*), $x^* := x$.
Step 3. *If* $x^* = x_0$ (no improvement found in step 2), stop;
else $x_0 := x^*$, and return to step 2. □

The second procedure solves (ROI-1) approximately by fixing the value of s found at x_0 at all the neighborhood points of x_0, and, due to its faster speed, is intended for experimentation on larger problem instances.

Local search 2 (LS-2)

Step 1. *Set* $x^* := x_0$.
Step 2. *Repeat* for each point in $\mathcal{N}_{loc}(x_0)$:
solve (P_2) for $s = s_f$ to obtain solution x';
if ROI$(x') >$ ROI (x^*), $x^* := x'$.
Step 3. *If* $x^* = x_0$, stop;
else solve (ROI-1) with algorithm 2 to upgrade incumbent solution x^*;
$x_0 := x^*$ ($s_f := s_f$ (new)), and return to step 2. □

Implementation details regarding the data structure used for swap moves, updating of the next solution, etc., are omitted here since they are adapted from Hansen and Mladenović (1997) or Hansen et al. (2003).

4 Using metaheuristics with embedded local search

Variable neighborhood search (VNS) is a recent metaheuristic for solving combinatorial and global optimization problems (e.g., Mladenović and Hansen, 1997, also see the surveys in Hansen and Mladenović, 2001, 2003), whose basic idea is to move the search systematically to different neighborhoods in order to "jump" out of a local optimum trap. Thus, in our case, the objective is to apply VNS in order to improve the incumbent solution obtained from the local search. To this end, a neighborhood structure must be defined on the solution space. Fortunately, we may use a similar neighborhood structure as given in Hansen et al. (2003) for the simple plant location problem (SPLP).

Let S denote any subset of open facilities, and $x = (Y, X, s)$ any optimal solution of (ROI-1) obtained for $S(= I_0)$, where Y is the vector of y_i values defined by S. Let $\mathcal{X}$ denote the space of all such possible solutions x. Note that the solution may not be unique for given S, so that the total number of points in $\mathcal{X}$ is bounded below by $2^m - 1$ (null set not included). Let $x_1(S_1), x_2(S_2)$, be any two solutions in $\mathcal{X}$; then we define the distance between them as

$$\rho(x_1, x_2) = |(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| = \|Y_1 - Y_2\|, \tag{12}$$

where the last term denotes the Hamming distance between Y_1 and Y_2. If $S_2 = S_1 \setminus \{i_1\}$ (a drop) or $S_2 = S_1 \cup \{i_2\}$ (an add), $\rho(x_1, x_2) = 1$; if $S_2 = S_1 \setminus \{i_1\} \cup \{i_2\}$ (an interchange), $\rho(x_1, x_2) = 2$. The k^{th} neighborhood $\mathcal{N}_k(x)$ of a current solution x is defined as the set of all possible solutions x' derived from x by any combination of exactly k total drop, add, or interchange moves. Note that the neighborhood used before for the local search ($\mathcal{N}_{loc}$) corresponds to $\mathcal{N}_1$ in the current formulation.

Using the neighborhood structure defined above, we are now ready to apply the steps of VNS on problem (ROI). The following parameters need to be specified or tailored for the problem instance at hand:

t_{max}: limit on execution time (stopping condition)

$(k_{min}, k_{max}, k_{step})$: respectively, the minimum and maximum neighborhood size and step between neighborhoods used in the shaking operation.

The basic version of VNS eliminates the requirement to customize parameters except for t_{max}, by setting $k_{min} = k_{step} = 1$, and $k_{max} = m$, i.e., the total solution space is covered. (Note that the maximum possible distance between two solutions is equal to m.) Once an initial solution $x(S)$ is found by local search, our implementation of VNS proceeds as follows.

Variable neighborhood search (VNS)

Repeat the following sequence until the time limit t_{max} is reached:

Step 1. $k := k_{min}$

Step 2. Until $k = k_{max}$, repeat the following steps:

(a) **Shaking**. Generate a subset S_k at random from the k^{th} neighborhood of the incumbent solution x, and solve (ROI-1) using algorithm 2 to obtain the associated point $x' \in \mathcal{N}_k(x)$.

(b) **Local search**. Apply one of the given variants of local search with x' as initial solution; denote by x'' the output solution.

(c) **Move or not**. If ROI(x'') > ROI(x), move there ($x := x''$), and continue the search with $\mathcal{N}_{k_{min}}$ ($k := k_{min}$); else, $k := k + k_{step}$. □

The shaking operation requires some further explanation. In order to obtain a random point x' in the k^{th} neighborhood of $x(S)$, drop, add, or interchange moves are chosen randomly k times while these moves are still available. Note that the maximum allowable number of drop moves = $\min\{k, |S|\}$ if $k \neq |S|$, and $k - 1$, if $k = |S|$, and maximum allowable number of add moves = $\min\{k, m - |S|\}$. Since the distance from x is increased by 2 for each interchange move, the maximum number of such moves = $\min\{k, m - |S|\}$, if $k \leq |S|$, and $\min\{|S|, m - k\}$, if $k > |S|$. The remaining open sites from S and closed sites from $I \setminus S$ are chosen randomly in each move depending on whether the move is a drop, add, or interchange. Further details of implementation follow closely the primal VNS procedure in Hansen et al. (2003) for the simple plant location problem.

Tabu search (TS) is a well–known metaheuristic that allows uphill (downhill) moves when a local minimum (maximum) is reached (see, e.g., Glover, 1989, 1990, Glover and Laguna, 1997). In our implementation of tabu search, we select a random starting point $x(S) \in \mathcal{X}$ and perform a local search (LS–1 or LS–2) starting from that point. The search is restricted to the $\mathcal{N}_1$ neighborhood of the current solution x, and a move is always made to the best solution $x' \in \mathcal{N}_1(x) \setminus T$, where $T \subset \mathcal{N}_1$ represents tabus or the set of forbidden moves.

Two tabu lists are maintained to avoid cycling, and we use the same stopping criterion (t_{max}) for comparison purposes with VNS. This is the single parameter used since the lengths of both tabu lists, that are usual TS parameters, are changed randomly in each iteration from [1, p] and [1, $m - p$] for the first and the second tabu list, respectively (p is the current number of open facilities). In fact, we apply the so-called *Chain interchange* TS suggested by Mladenović et al. (1996), where facilities currently in the solution (V_{in}) and facilities out (V_{out}) are both divided into non-tabu and tabu subsets (V_{in}^n, V_{in}^t, V_{out}^n, V_{out}^t). The interchange of two facilities is performed by changing positions of four elements in the list: the facility that goes out is chosen from V_{in}^n and replaced with one that belongs to the solution as well (i.e., from V_{in}^t), but whose turn has come to change tabu status; in its place comes a facility from V_{out}^n, which is substituted by an element from V_{out}^t; finally, the chain is closed by placing the facility from V_{in}^n in V_{out}^t. In that way the Tabu Search *recency based memory* is easily exploited, i.e., the possibility of getting out of a local optimum is achieved without additional efforts. For add or drop moves, the corresponding tabu list is taken into account.

Chain-interchange Tabu search *(TS)*

Step 1*: Initialization:*

(1) Find initial solution x; set $x^* := x$;
(2) Initialize Tabu lists ($V_{in}^t = V_{out}^t = \emptyset$).

Step 2*: Repeat* until the time limit t_{max} is reached:

(1) Find the best non-tabu solution x' in the $\mathcal{N}_1$ neighborhood of x;
(2) If it is improved, then save that solution ($x^* := x'$);
(3) Update:
  (i) the current solution ($x := x'$);
  (ii) the Tabu lists;
  (iii) the first and the second most profitable open facilities of each client. □

The details regarding implementation are again avoided, since the procedure is very similar to the one already given in Mladenović et al. (2003), where Chain-interchange TS is applied to the p-center problem.

5 Computational results

Random test instances are generated as follows: (1) the set of potential facilities coincides with the set of customers (m=n); the coordinates of customers/ potential facilities are generated by a uniform distribution over a [0,200] × [0,200] square; (2) Euclidean distances between customers and facilities make up the transportation cost matrix $C = [c_{ij}]$; (3) $e_i = 1$, for all i; (4) $f_i = \lfloor 45 \cdot \sqrt{n} \rfloor$; (5) $g_i = 200 \cdot \sqrt{2} \cdot 0.5 \cdot rand(0,1)$; (6) $d_j = \max_{i=1,\ldots,m} c_{ij}/3$; (7) $p_j = d_j \cdot 1.5 \cdot rand(0,1)$.

The test instances generated using these parameter settings are intended only to allow a preliminary comparison of solution quality and running times of the different heuristics proposed, and not to simulate practical problems or a range of difficult test problems.

Local search. An empirical comparison between the two local search procedures is given in Table 1. The first two columns give the remaining parameters (m, α) of the problem instances, while the next two columns report average (out of 10 randomly generated test instances) objective values obtained by LS-1 and LS-2 respectively. The % deviation is calculated as

$$\frac{z(LS_1) - z(LS_2)}{z(LS_1)} \cdot 100. \tag{13}$$

The last two columns give corresponding average computing times in seconds. Observe that LS-1 is 0.60% better on average than LS-2, but several times slower. Hence, we opted to use LS-2 in all heuristics subsequently tested.

Comparison of heuristics. In Table 2, results obtained by our VNS and TS for solving ROI with market share constraint are presented. A multistart local search (MLS) procedure is also included. The columns have the same meaning as in Table 1. The % deviation is now calculated as $(z_{best} - z_{heur})/z_{best} \cdot 100$. Each line reports the average values out of 10 randomly generated test instances with the same (m, α). Since ROI is a maximization problem, a larger value gives a better result. It appears that the VNS and TS methods are of similar quality, although VNS is slightly better. This is indicated by non-negative values of the % dev of TS for all test instances. The MLS procedure performs on average the worst, although it does obtain better results than TS on some of the cases.

		Obj.value			Time	
m	α	LS–1	LS–2	% dev	LS–1	LS–2
200	0.90	103.67	102.60	1.03	4.09	0.32
	0.80	114.58	113.93	0.57	3.37	0.34
	0.70	123.45	123.09	0.29	2.75	0.29
	0.60	133.56	133.13	0.32	2.47	0.31
	0.50	142.15	140.56	1.12	2.00	0.24
	0.40	150.73	149.18	1.03	2.64	0.26
	0.30	158.14	157.56	0.37	2.34	0.28
	0.20	159.93	157.70	1.39	2.24	0.32
	0.10	162.75	160.89	1.14	2.52	0.37
500	0.90	114.75	114.05	0.61	140.69	2.34
	0.80	126.65	125.84	0.64	134.32	2.38
	0.70	137.70	136.88	0.60	95.48	2.48
	0.60	148.75	148.35	0.27	68.19	1.71
	0.50	157.86	157.53	0.21	73.96	2.01
	0.40	167.30	167.29	0.01	59.67	1.89
	0.30	174.87	174.34	0.30	56.69	1.86
	0.20	179.48	179.14	0.19	69.72	3.29
	0.10	179.88	175.96	2.18	70.97	4.18
1000	0.90	120.41	120.02	0.32	972.83	9.08
	0.80	133.44	132.73	0.53	770.42	9.73
	0.70	145.94	145.17	0.53	770.76	11.55
	0.60	157.76	157.03	0.46	711.11	8.29
	0.50	168.64	168.22	0.25	451.38	10.09
	0.40	178.38	177.98	0.22	482.12	8.51
	0.30	187.58	187.16	0.22	502.71	10.92
	0.20	193.27	193.12	0.08	567.12	13.64
	0.10	197.71	195.34	1.20	505.80	13.73
Aver.		152.57	151.66	0.60	241.79	4.46

Table 1: Comparison of two local search procedures.

m	α	Obj. value VNS	Obj. value TS	Obj. value MLS	% deviation VNS	% deviation TS	% deviation MLS	Time VNS	Time TS	Time MLS
200	0.90	104.90	104.83	104.29	0.00	0.07	0.58	14.74	9.64	18.52
	0.80	115.44	115.19	115.03	0.00	0.22	0.36	34.72	11.15	13.56
	0.70	125.71	125.45	124.58	0.00	0.21	0.90	26.66	10.38	5.44
	0.60	135.21	135.17	134.38	0.00	0.03	0.61	10.59	9.89	2.97
	0.50	144.08	143.99	143.12	0.00	0.06	0.67	8.47	9.97	1.49
	0.40	152.35	152.12	151.54	0.00	0.15	0.53	9.76	8.15	2.12
	0.30	158.71	158.71	158.42	0.00	0.00	0.18	28.61	7.49	0.94
	0.20	162.32	162.32	161.12	0.00	0.00	0.74	4.32	2.87	1.42
	0.10	164.61	164.61	163.68	0.00	0.00	0.56	2.03	0.92	1.27
500	0.90	115.72	115.38	115.20	0.00	0.29	0.45	42.11	70.69	120.64
	0.80	127.77	127.33	127.18	0.00	0.34	0.46	17.97	104.77	152.57
	0.70	139.19	138.72	138.45	0.00	0.34	0.53	55.33	130.82	40.02
	0.60	149.71	149.39	149.23	0.00	0.21	0.32	43.24	80.71	38.02
	0.50	159.65	159.16	158.75	0.00	0.31	0.56	44.09	92.95	50.76
	0.40	168.29	167.46	167.79	0.00	0.49	0.30	119.96	37.19	31.72
	0.30	175.78	175.73	175.41	0.00	0.03	0.21	44.34	154.68	12.83
	0.20	180.24	180.22	179.87	0.00	0.01	0.21	85.80	79.23	5.49
	0.10	183.80	183.80	181.84	0.00	0.00	1.07	15.62	18.37	5.10
1000	0.90	121.91	121.28	121.06	0.00	0.52	0.70	122.13	206.48	313.31
	0.80	134.55	134.21	133.92	0.00	0.25	0.47	57.97	136.60	291.79
	0.70	146.98	146.19	146.45	0.00	0.54	0.36	192.00	278.75	279.93
	0.60	159.00	158.08	158.34	0.00	0.58	0.42	78.99	144.42	230.55
	0.50	170.13	169.47	169.37	0.00	0.39	0.45	151.49	163.65	144.30
	0.40	180.12	178.79	179.25	0.00	0.74	0.48	241.17	139.63	133.06
	0.30	187.66	186.86	187.44	0.00	0.43	0.12	126.70	158.16	52.40
	0.20	194.13	194.07	193.85	0.00	0.03	0.14	197.37	176.03	22.74
	0.10	200.08	200.08	198.90	0.00	0.00	0.59	43.47	72.74	72.32
Aver.		154.00	153.65	153.28	0.00	0.23	0.48	67.39	85.79	75.75

Table 2: Comparison of VNS, TS and MLS.

6 Conclusions

In this paper, we present a plant location problem which maximizes return on investment subject to a lower bound on market share. This model is an important extension of the much-studied simple plant location problem. We give an exact solution procedure when the market share constraint is relaxed. The model also is amenable to exact solution when the subset of open facilities is specified. Building on this, we develop a local search procedure for the problem, and then use this procedure within a variable neighborhood search and tabu search heuristic. Preliminary computational experience is given. These heuristics allow realistic problem sizes to be investigated, and to our best knowledge, represent the first attempt to use heuristics on the maximum return-on-investment plant location problem. Areas for future research include developing other heuristics, and doing extensive computational experiments to refine the parameter settings.

References

[1] Balinski, M. (1965). Integer programming: methods, uses, computation. *Management Sci.* **12**: 254–313.

[2] Stollsteimer, J.F. (1963). A working model for plant numbers and locations. *J. Farm. Econom.*: 631–645.

[3] Kuehn, A.A. and Hamburger, M.J. (1963). A heuristic program for locating warehouses. *Management Sci.* **9**: 643-666.

[4] Krarup, J. and Pruzan, P.M. (1983). The simple plant location problem: Survey and synthesis. *Eur. J. Opl. Res.* **12**: 36–81.

[5] Labbé, M., Peeters, D., and Thisse, J.F. (1995). Location on networks. Ball, M., Magnanti, T., Monma, C., and Nemhauser, G. (eds.), *Handbook of Operations Research and Management Science: Networks*. North Holland, Amsterdam.

[6] Labbé, M. and Louveaux, F.V. (1997). Location problem. DellAmico, N., Maffioli, F., Martello, S. (eds.), *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, New York, 264-271.

[7] Mirchandani, P.B. and Francis, R.L. (1990). *Discrete location theory*. Wiley-Interscience, New York.

[8] Francis, R.L., McGinnis Jr., L.F., and White, J.A. (1992). *Facility layout and location: an analytical approach*. 2nd Edn International Series in Industrial and Systems Engineering. Prentice Hall, Englewood Cliffs, N.J.

[9] Daskin, M. (1995). *Network and Discrete Location: Models, Algorithms and Applications*. Wiley, New York.

[10] ReVelle, C. and Laporte, G. (1996). The plant location problem: new models and research prospects. *Oper. Res.* **44**: 864–874.

[11] Myung, V.S., Kim, H., and Tcha, D. (1997). A bi-objective uncapacitated facility location problem. *Eur. J. Opl. Res.* **100**: 608-616.

[12] Brimberg, J. and ReVelle, C. (2000). The maximum return-on-investment plant location problem. *J. Opl. Res. Soc.* **51**: 729–735.

[13] Brimberg, J. and ReVelle, C. (1998). A bi-objective plant location problem: cost vs. demand served. *Location Sci.* **6**: 121–135.

[14] Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science* **13**: 492–498.

[15] Manne, A.S. (1964). Plant location under economies-of-scale-decentralization and computation. *Management Sci.* **11**: 213-235.

[16] Feldman, E., Lehrer, F.A., and Ray, T.L. (1966). Warehouse Location under Continuous Economies of Scale. *Management Science* **12**: 670–684.

[17] Teitz, M.B. and Bart, P. (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. *Oper. Res.* **16**: 955–961.

[18] Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. (2001). Local search heuristics for k-median and facility location problems. *ACM Symposium on Theory of Computing*: 21–29.

[19] Ghosh, D. (2003). Neighborhood search heuristics for the uncapacitated facility location problem. *Eur. J. Opl. Res.* **150**: 150–162.

[20] Hansen, P., Brimberg, J., Urosevic, D., and Mladenovic, N. (2003). Primal-dual variable neighborhood search for bounded heuristic and exact solution of the simple plant location problem. *Les Cahiers du GERAD* G–2003–64. HEC Montréal.

[21] Hansen, P. and Mladenović, N. (1997). Variable neighborhood search for the p-Median. *Location Sci.* **5**: 207–226.

[22] Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers Oper. Res.* 24: 1097–1100.

[23] Hansen, P. and Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European J. of Oper. Res.* **130**: 449–467.

[24] Hansen, P. and Mladenovic, N. (2003). Variable Neighborhood Search. Glover, F. and Kochenberger, G. (eds.), *Handbook of Metaheuristics*, 145–184. Kluwer Academic Publisher.

[25] Glover, F. (1989). Tabu search. Part I. *ORSA J. Comput.* **1**: 190–206.

[26] Glover, F. (1990). Tabu search. Part II. *ORSA J. Comput.* **2**: 4–32.

[27] Glover, F. and Laguna, M. (1997). *Tabu Search.* Kluwer Academic Publishers, Norwell, MA.

[28] Mladenović, N., Moreno-Perez, J.A., and Moreno-Vega, J. (1996). A chain-interchange heuristic method. *Yugoslav J. Opns. Res.*: 41–54.

[29] Mladenović, N., Labbé, M., and Hansen, P. (2003). Solving the p-center problem by tabu search and variable neighborhood search. *Networks* 42: 48–64.