Budget Constrained Location Problem with Opening and Closing of Facilities

By: Qian Wang, Rajan Batta, Joyendu Bhadury, and Christopher M. Rump

J. Bhadury, Q. Wang, R. Batta and C. Rump. "Budget Constrained Location problems With Opening and Closing Of Facilities", *Computers and Operations Research*, Vol. 30, 2047-2069 (2003)
doi:10.1016/S0305-0548(02)00123-5

Abstract:
In this paper, we study a budget constrained location problem in which we simultaneously consider opening some new facilities and closing some existing facilities. Motivations for this problem stem from applications where, due to a change in the distribution of customer demand, the existing facility system no longer provides adequate service. The objective is to minimize the total weighted travel distance for customers subject to a constraint on the budget for opening and/or closing facilities and a constraint on the total number of open facilities desired. For this problem, we develop a mathematical programming model and examine its theoretical properties. We then develop three heuristic algorithms (greedy interchange, tabu search and Lagrangian relaxation approximation) for this NP-hard problem. Computational testing of these algorithms includes an analysis of the sensitivity of the solution to the budget and the desired number of facilities. The intended application in this testing is that of locating/relocating bank branches in a large-size town such as in our data set from Amherst, New York. We also discuss the situation where operating costs are part of the objective function.

Scope and purpose
In many service industry applications, e.g., banking, changing customer demand renders the current set of facilities inefficient. Under a budget constraint, a company may wish to close some of its existing facilities and open some new facilities to better serve its customers. This paper studies such a situation, drawing upon the literature for related location models and their solution techniques. An example of the scenario of locating bank branches in a large township is presented.

Keywords: Facility location; Greedy algorithm; Lagrangian relaxation; Tabu search

1. Introduction
Facility location problems commonly occur in the following form: a number of facilities are going to be located in a region to satisfy the demand of customers. It is required to select the optimal set of locations for the facilities among the potential location sites according to a certain objective function.

Our work is related to the p-median literature, which is concerned with selecting sites for p new facilities so as to minimize the average travel distance of customers, assuming that customers use the closest facility. For this problem, Hakimi [1] proved the node optimality property which says that at least one optimal solution consists of locating p facilities on the network's nodes. Levy [2] and Handler and Mirchandani [3] dealt with a concave cost function for this problem. Kariv and Hakimi [4] established that the p-median on a general network is NP-hard. However, several solution procedures have been tested with varying degrees of success. Notable amongst these are: (i) a greedy approach by Cornuejols et al. [5], (ii) node partitioning by Maranzana [6], (iii) node substitution by Teitz and Bart [7], (iv) heuristic branch-and-bound by Jarvinen et al. [8], (v) a primal and dual heuristic by Captivo [9], and (vi) meta-heuristic techniques, such as a binomic approach by Maniezzo et al. [10] and a gamma heuristic (heuristic concentration) by Rosing et al. [11].

Our work is also related to the uncapacitated facility location problem (UFLP)—see, Cornuejols et al. [12] for a survey. The UFLP considers opening new facilities to satisfy the demand of customers when there are currently no existing facilities. It accounts for both the fixed cost of opening the facilities as well as the travel cost for customers, who are assumed to use the closest facility. Many algorithms have been developed to solve this problem. Heuristics for the UFLP include: (i) the greedy and interchange heuristic by Kuehn and Hamburger [13] and Cornuejols et al. [5], and (ii) the dual descent heuristic by Erlenkotter [14] and Tcha et al. [15]. Based on the mixed-integer programming formulation for the UFLP developed by Balinski [16], many mathematical programming approaches for the UFLP have been proposed. A branch-and-bound scheme is widely used among the papers in this area. Recently, some researchers have also studied large-scale problems and problems with special structure (see, for example, Denshan and Rushton [17] and Jones et al. [18]).

Among all the papers on the p-median and the UFLP, only a few considered the problem of opening new facilities or closing existing facilities when some facilities already exist. Berman and Simchi-Levi [19] and Drezner [20] considered the problem of adding some new facilities. Chhajed and Lowe [21] studied the problem of adding m new facilities on a tree, given that there are n pre-existing facilities. They found an $O(mn)$ algorithm for this problem. Leorch et al. [22] considered the problem of closing some existing facilities. Dell [23] focused on the formulation of closing or realigning of US Army installations. Savage et al. (see Associated Press [24]) studied the relocation of ATMs for Wells Fargo Bank.

The problem of relocating a facility can be viewed also a highly application specific situation. Once such study is that by Min and Melachrinoudis [25] in which they present a real-world case study for the relocation of a combined manufacturing and distribution (warehouse) facility. They include the six main criteria in a real business case—site characteristics, cost, traffic access, market opportunity, quality of living, and local incentives. A particular highlight of their work is the consideration given to the design of supply chain network in the relocation decision. In a later paper they built a math programming model for the same problem, see Melachrinoudis and Min [26]. Our work does not have such a specific application in mind. However, the intended application for our model is that of locating bank branches in an urban environment. For this reason, we have done our computational testing on a realistic network of a major town in New York State. We have also chosen our model parameters in accordance with this application setting.

From a modelling perspective, the primary point of departure in this paper is the tenet that is frequently followed in real-life applications: under certain circumstances it is unrealistic to consider the issue of opening new facilities or closing existing ones in isolation. We submit that in contrast, many real applications have to consider both issues simultaneously. Consider the example of a bank that already has several branches spread over a town. Due to the change of population distribution, some customers may complain that there are no branches nearby while only a few customers visit some of the existing branches. Therefore, the management of the bank decides to invest a certain amount of their financial budget to relocate some of the branches so that they can improve their service by decreasing the customers' total travel cost (distance). The budget is enough to open a number of new branches, but the management does not want their total number of branches, including both new and old, to exceed a certain number due to an upper limit on the annual operating costs. Hence, the problem that the management faces is to decide where to open new branches and which existing branches should be closed so that the total travel distance for customers is minimized, subject to the constraints on the budget and the maximum number of facilities.

The significance of our work is as follows: It is first to consider and analyze a model specifically designed for simultaneously opening and closing of facilities. Second, it successfully adapts solution techniques from the literature for the p-median problem and the UFLP problems, to analyze the issues of fixed costs of opening/closing of facilities and travel costs for serving customers, who are assumed to be assigned to their closest facility. As a final point, our methods can also be tailored to handle operating costs for facilities.

The rest of the paper is organized as follows. The next section contains a mathematical formulation of the problem. Some model properties and insights are presented in Section 3. Section 4 develops three heuristics for the problem. Section 5 presents our computational results and sensitivity analysis on budget and maximum number of facilities. Extensions to the situation where operating cost is part of the objective function are discussed in Section 6. Finally, Section 7 contains a summary, our conclusions and some ideas for future research.

2. Formulation

Consider a system which can be represented by a network $G(M, N_1, N_2, D)$, where

$M = \{1,\ldots,m\}$: the set of nodes where customers reside,
$N = \{1,\ldots,n\} = N_1 \cup N_2$: the set of all facility nodes, where
$N_1 = \{j_1,\ldots,j_q\}$: the set of nodes where existing facilities are located,
$N_2 = \{j_{q+1},\ldots,j_n\}$: the set of nodes where new facilities can be located,
D: the distance matrix between customer nodes and facility nodes,
d_{ij}: the distance between customer node $i \in M$ and facility node $j \in N$,
w_i: the demand (weight) of customer node $i \in M$,
p: the total number of open facilities desired by the management,
$q = |N_1|$: the number of existing facilities,
$n = |N|$: the total number of existing and potential facility sites,
c_j: the cost to close an existing facility at node $j \in N_1$ or open a new facility at node $j \in N_2$,
b: the total budget available.

Note that the customer nodes M and facility nodes N are not necessarily disjoint, i.e., they may share common nodes.

The facility relocation problem is to find the sites of the existing facilities to close and the sites for new facilities to open so that the total weighted travel distance is minimized, subject to the constraints on the budget and the total number of open facilities desired.

Similar to the UFLP, the problem described above has two parts: (1) selecting nodes in N which will have a facility (if a node in N_1 is not selected, then the existing facility at the node should be closed; if a node in N_2 is selected, then a new facility should be opened at the node), and (2) assigning nodes in M to the closest open facility.

Define

$$y_j = \begin{cases} 1 & \text{if facility } j \in N \text{ is open,} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1 & \text{if demand of customer } i \in M \text{ is satisfied by the facility opened at } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

If $y_j = 0$ for $j \in N_1$, then the existing facility $j \in N_1$ is going to be closed, incurring a cost of $c_j(1 - y_j)$; if $y_j = 1$ for $j \in N_2$, then a new facility is going to be opened at potential facility site j, incurring a cost of $c_j y_j$. Since the total budget available is b, we have

$$\sum_{j \in N_1} c_j(1 - y_j) + \sum_{j \in N_2} c_j y_j \le b.$$

Since p facilities are desired, $\sum_{j \in N} y_j = p$. Since a customer can only be assigned to an open facility site, we have $x_{ij} \leq y_j$, $\forall i \in M, j \in N$. Every customer is assigned to exactly one location (ties are broken arbitrarily), so we must have $\sum_{j \in N} x_{ij} = 1$; $\forall i \in M$. Therefore, the problem can be formulated as follows:

$$\text{(P)Minimize} \quad \sum_{i \in M} \sum_{j \in N} w_i d_{ij} x_{ij}$$

subject to the constraints :

$$\sum_{j \in N_1} c_j (1 - y_j) + \sum_{j \in N_2} c_j y_j \leq b. \tag{1}$$

$$\sum_{j \in N_1} x_{ij} = 1 \quad \forall i \in M, \tag{2}$$

$$\sum_{j \in N_1} y_j = p, \tag{3}$$

$$x_{ij} \leq y_j \ \forall i \in \mathrm{M}, j \in N, \tag{4}$$

$$y_j \in \{0,1\} \ \forall j \in N, \tag{5}$$

$$x_{ij} \geq 0 \ \forall i \in M, j \in N. \tag{6}$$

We note that y_j being binary will force x_{ij} to be binary. Hence, in constraint (6) we let x_{ij} be a continuous variable.

3. Model properties and insights

We now establish some basic properties for problem (P) for insights into the model.

3.1. Relation to the vertex covering problem

The *p*-median problem is a special case of problem (*P*) when $N_1 = \emptyset$ and $b = +\infty$. Since the p-median problem is NP-hard (Kariv and Hakimi [4]), it follows that (*P*) is NP-hard.

Another interesting feature is that the vertex covering problem, which is NP-complete, can be transformed into the problem (*P*). To see this, consider the formal statement of vertex covering problem: Given a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, and an integer $K \leq |V|$, the vertex cover problem is to find a subset $V' \subset V$ such that $|V'| \leq K$ and, for each edge $(u, v) \in E$, at least one of u and v belongs to V'. We now transform this problem into an instance of (*P*) by setting: $M = E$, $N_1 = \emptyset$, $N = N_2 = V$, $b = |V|$, $q = |N_1| = 0$, $p = K$, $w_i = 1$, $\forall i \in M$, $c_j = 1$, $\forall j \in N$,

$$and\ d_{ij} = \begin{cases} 1 & \text{if vertex } j \text{ is adjacent to edge } i, \\ 2 & \text{otherwise.} \end{cases}$$

This instance of (*P*) tries to find a cover of all edges with p vertices in V. We observe that since the budget is sufficient to open K facilities, this instance of (*P*) is always feasible. The vertex cover problem is feasible if and only if the optimal solution of this instance of (*P*) has an objective function value of $|E|$. In other words, if the vertex cover problem is infeasible, then the optimal objective function value of the (*P*) is greater than $|E|$, and vice versa. Therefore, an optimal solution of this instance of (*P*) with an objective function value of $|E|$ provides a feasible solution of the corresponding vertex cover problem. We note that this transformation is polynomial in the size of graph G. The benefit of this property is that the heuristic algorithms developed and tested for (P) can also be used to solve the vertex cover problem.

3.2. Budgetary range

To determine the largest value of the budget parameter b, we drop constraint (1) and find the solution of the resultant problem relaxation. This is a p-median problem in which N constitutes the set of facility nodes. Let $b_{\max}$ be the resulting sum of the opening and closing costs. This is an upper bound on the budget required to solve this relocation problem.

Finding the minimum budget value, $b_{\min}$, is even easier. If $q = p$, then $b_{\min} = 0$, since we can keep the current set of facilities. If $q > p$, then $b_{\min}$ is the summation of the smallest $q — p$ closing costs in the facility set N_1. That is, if there are currently more open facilities than desired, the least expensive thing to do is to close those $q — p$ facilities with the smallest closing costs. Finally, if $q < p$, then $b_{\min}$ is the summation of the smallest $p — q$ opening costs in the facility set N_2. That is, if there are currently fewer open facilities than desired, the least expensive thing to do is to open those $p — q$ facilities with the smallest opening costs.

When $b_{\max} — b_{\min}$ is large in comparison to $b_{\max}$, intermediate values of b (those in the middle of the range) would allow for a combinatorially large number of combinations of facilities to close/open. The problem is therefore more difficult for these values of b. It is also these values of b that make the problem formulation more relevant since the closing of existing facilities and the opening of new ones occur simultaneously.

3.3. Relaxation on the number of desired facilities

We now relax constraint (3) as

$$\sum_{j \in N} y_j \geq p,$$

and denote the relaxed problem as (P_{relax}). The following propositions provide sufficient conditions under which an optimal solution to the relaxed problem (P_{relax}) also optimally solves problem (P). In other words, under these conditions, the lower bound for problem (P) obtained by solving the relaxation (P_{relax}) will be tight.

Proposition 3.1. *If $q \geq p$, then there exists an optimal solution to* (P_{relax}) *that has p open facilities.*

Proof. Suppose there exists an optimal solution S to (P_{relax}) that has less than p open facilities. The following procedure shows that we can generate a solution from S that has p open facilities. Since $q > p$, in order to reach S, at least $q — p + 1$ facilities should be closed. Reopening any one of the closed facilities, say facility j, cannot degrade the objective value, i.e., the objective value of $S^1 = S \cup j$ would be no worse than the one of S. It is easy to see that S^1 is a feasible solution to (P_{relax}) since the budget S^1 used is less than that of S (one less facility is closed), and the number of open facilities is less than or equal to p. Therefore, it is also an optimal solution. If $|S^1| = |S| + 1 = p$, then we have obtained an optimal solution with p open facilities. Otherwise, we can continue this procedure to generate an optimal solution that has exactly p open facilities. ■

Proposition 3.2. *If* q < p, *then let $\bar{b}$ denote the summation of the largest p — q opening costs c_j. If $b \geq \bar{b}$, then there exists an optimal solution to* (P_{relax}) *that has p open facilities.*

Proof. Suppose there exists an optimal solution S to (P_{relax}) that has less than p open facilities. Two cases may occur: (i) $N_1 \subset S$, which means that no existing facilities are closed in order to get S or (ii) some existing facilities are closed in order to get S. The following shows that in both cases we can generate a solution from S that has p open facilities.

Case (i): Since no budget has been used to close any existing facilities in S and $b \geq \bar{b}$, we still have budget to open any potential facility $j \in N \setminus S$. The objective value of $S^1 = S \cup j$ would not be worse than the one of S. It is easy to see that S^1 is a feasible solution to (P_{relax}) since the budget S^1 used is less than $\bar{b} \leq b$, and the number of open facilities is less than or equal to p. Therefore, it is also an optimal solution.

Case (ii): Reopening any one of the closed facilities, say facility j, cannot degrade the objective value, i.e., the objective value of $S^1 = S \cup j$ would be no worse than the one of S. It is easy to see that S^1 is a feasible solution to (P_{relax}) since the budget S^1 used is less than that of S (one less facility is closed), and the number of open facilities is less than or equal to p. Therefore, it is also an optimal solution.

If $|S^1| = |S| + 1 < p$, then we have obtained an optimal solution with p open facilities. Otherwise, S^1 must lie either in case (i) or case (ii). We can continue this procedure to generate an optimal solution that has exactly p open facilities.

From the above propositions, we can conclude that the heuristics developed for (P) in this paper can also be applied to most instances of (P_{relax}).

4. Heuristic algorithms

In this section, three heuristic approaches are developed for our problem. The first one is a greedy-interchange (GI) heuristic, the second one uses a tabu search (TS) technique, and the third one is based upon Lagrangian relaxation (LR).

4.1. Greedy-interchange (GI) heuristic

This algorithm consists of two phases. In phase one we either open or close facilities to make the total number of open facilities equal to p. Let Q be the set of facility nodes which are open, $C(Q)$ be the total of opening and closing costs to achieve open facilities only on the set Q and $\delta_i = \min\{d_{il}: l \in Q\}$ be the travel distance to set Q for the customers at $i \in M$.

If $q < p$, we need more facilities. If the summation of the smallest $(p-q)$ opening costs is greater than b, which means that the budget is not enough to open $(p - q)$ new facilities, then the problem is infeasible. Otherwise, the problem is feasible. To find a feasible solution, we sequentially add new facilities until there are p open facilities. If a new facility is added, the total weighted travel distance will either remain the same or decrease. The amount of decrease after adding a facility at node $j \in N_2$ can be evaluated by calculating

$$\phi_j = \sum_{i \in M} w_i \max(0, \delta_i - d_{ij})$$

The criterion for selecting a facility to open at each step is to select the one with the largest benefit-to-cost ratio ϕ_j/c_j from those nodes satisfying $C(Q \cup \{j\}) < b$. If there is no such node available, then we deselect the facility with largest opening cost among the newly selected facilities to open, and instead select to open the facility with the smallest opening cost among those not yet selected. The procedure is continued until p facilities are opened.

If $q > p$, we need to close some facilities. Similar to the case of opening facilities, if the summation of the smallest $(q - p)$ closing costs is greater than b, which means that the budget is not enough to close $(q - p)$ existing facilities, then the problem is infeasible. Otherwise, the problem is feasible. To find a feasible solution, we sequentially close existing facilities until p are left. If an existing facility is closed, the total weighted travel distance will either be unchanged or increase. The amount of increase after closing a facility at site j is obtained by the expression

$$\phi_j = \sum_{i \in M} w_i \min\{d_{ik} - \delta_i\} : k \in Q \setminus \{j\}\}.$$

The criterion for selecting an existing facility to close is to choose the one with the smallest value of $\phi_j = c_j$ from those nodes satisfying $C(Q \setminus \{j\}) < b$. If there is no such node available, then we deselect the facility with the largest closing cost among the newly selected facilities to close, and instead select the one with the smallest closing cost among those not yet selected. The procedure is continued until p facilities remain.

Phase two is an interchange procedure that performs a swap between a pair of facility nodes. The pair should consist of a node $j \in N$ with an open facility and a node $k \in N$ without an open facility. If there exist feasible swaps that improve the objective function value, the one with the best objective function value is selected. (Here feasible means that the cost after opening a facility at k and closing a facility at j is within budget.) The interchange procedure continues until either there are no feasible swaps or no feasible swaps improve the objective function value. A detailed presentation of this GI algorithm is provided in Wang [27].

Example 1. Consider a network with $m = 8$ customer nodes and $n = 5$ facility nodes, with weighted distance matrix

$$w_i d_{ij} = \begin{pmatrix} 10 & 32 & 18 & 25 & 34 \\ 13 & 25 & 9 & 27 & 33 \\ 19 & 24 & 24 & 9 & 18 \\ 6 & 16 & 7 & 12 & 19 \\ 19 & 5 & 14 & 9 & 8 \\ 27 & 17 & 17 & 29 & 28 \\ 32 & 10 & 28 & 18 & 11 \\ 33 & 23 & 32 & 15 & 12 \end{pmatrix},$$

opening/closing costs c = (25, 23, 7, 5, 22) and a budget $b = 55$. Suppose further that facilities 3 and 4 are currently open ($N_1 = \{3;4\}$; $q = |N_1| = 2$), and a total of $p = 3$ open facilities are desired.

Iteration 1: Initially, let the existing facility set be the open facility set, $Q^1 = N_1 = \{3;4\}$. At this time, the weighted travel distance from each customer to his closest facility is $u^1 = (18, 9, 9, 7, 9, 17, 18, 15)$, the total weighted travel distance is $f(Q^1) = 102$, and the cost for closing/opening is $C = 0$.

Adding a facility at node 1, 2 or 5 will decrease the objective function value by $\phi_1 = 9$; $\phi_2 = 12$ and $\phi_5 = 11$, respectively. Since $\frac{12}{23}$ has the maximal value of ϕ_j/c_j, node 2 is selected, and the new set of facilities is given by $Q^2 = \{2, 3, 4\}$. After this, the travel distance from each customer to his/her closest facility is recalculated to be $u^2 = (18, 9, 9, 7, 5, 17, 10, 15)$, the total weighted travel distance is $f(Q^2) = f(Q^1) - \phi_2 = 90$ and the cost incurred for closing/opening is $C = 23$. Since we now have the desired three open facilities, we perform swaps between the open and non-open sites.

Iteration 2: Among the possible swaps, the one between facilities 1 and 3 produces the best objective function value, and we therefore obtain $Q^3 = \{1, 2, 4\}$, $f(Q^3) = 85$, $C = 55$. Since no further swap can improve the objective function, we stop. We conclude that the best solution is to open new facilities at nodes 1 and 2 and to close the existing facility at node 3.

4.2. Tabu search (TS) approach

In the interchange phase of the GI heuristic, the algorithm stops when no swaps improve the objective function value. This means that it stops at the first local optimum that it reaches. In order to have an opportunity to investigate the other local optima after reaching the first one, we modify the interchange phase by applying a TS technique; see Glover and Laguna [28] for a general description of the TS technique.

Pairwise swaps are still defined as the neighborhood of the tabu search. At each iteration, instead of terminating when no swap improves the objective function value, the best feasible swap is implemented, even if this results in a larger objective value. The algorithm continues until K successive swaps show no local improvement on the objective function. Furthermore, a tabu list of size L is generated. The tabu list contains a list of the L most recent swaps that did not locally improve the objective value. This prevents the search from repeating swaps tried in the recent past. A detailed presentation of this TS algorithm is provided in Wang [27].

Since the TS approach keeps searching other local optima after reaching the first local optimum, while the GI heuristic stops when it reaches the first local optimum (the GI heuristic is a sub-procedure in the TS approach), the TS approach will always take more time than the GI heuristic, and it will always provide no worse a solution than the GI heuristic.

The TS approach does not typically yield an improvement for small problems (like our example). It does, however, improve solutions for some large-scale problems as reported in Section 5.4.

4.3. Lagrangian relaxation (LR) approximation

Both the GI heuristic and the TS approach cannot guarantee the quality of the solution that they generate. Some researchers have developed solution procedures for the p-median problem and the facility location problem based on Lagrangian relaxation, see Narula et al. [29] and Cornuejols et al. [5]. In this subsection, we introduce a heuristic method based on LR that will generate an ε-optimal solution. A more detailed description of Lagrangian relaxation can be found in Beasley [30].

Our definition of ε-optimal is given by the following.

Definition. If z is the objective function value of a solution x and z^* is the optimal objective function value, x is said to be an ε-optimal solution if $(z - z^*)/z^* \leq \varepsilon$.

First, we consider a Lagrangian relaxation of problem (P). The LR $L(u; v)$ of problem (P) with respect to constraints (1) and (2) for a given (u; v) is

$$L(u,v) = \min_{x,y} \sum_{i \in M} \sum_{j \in N} w_i d_{ij} x_{ij} + \sum_{i \in M} u_i \left(1 - \sum_{j \in N} x_{ij}\right) + v \left[\sum_{j \in N_1} c_j (1 - y_j) + \sum_{j \in N_2} c_j y_j - b\right],$$

subject to constraints (3)–(6):

The objective function in the relaxation $L(u; v)$ can be rewritten as

$$\sum_{i \in M} \sum_{j \in N} (w_i d_{ij} - u_i) x_{ij} - \sum_{j \in N_1} v c_j y_j + \sum_{j \in N_2} v c_j y_j + \sum_{i \in M} u_i + v \sum_{j \in N_1} c_j - vb. \tag{7}$$

Define $(w_i d_{ij} - u_i)^- = \min\{0, w_i d_{ij} - u_i\}$, and let

$$\rho_j(u,v) = \begin{cases} \sum_{i \in M} (w_i d_{ij} - u_i)^- - v c_j & \text{if } j \in N_1, \\ \sum_{i \in M} (w_i d_{ij} - u_i)^- + v c_j & \text{if } j \in N_2. \end{cases}$$

Denote the smallest p such as p_j's by $p_{j_1}, p_{j_2},\ldots, p_{j_p}$. Since the last three terms in (7) are constant, it follows by inspection that the optimal solution to the LR for a given u and v is

$$y_j = \begin{cases} 1 & \text{if } j \in \{j_1, j_2, \ldots, j_p \\ 0 & \text{otherwise;} \end{cases} \tag{8}$$

$$x_{ij} = \begin{cases} y_j & \text{if } w_i d_{ij} - u_i < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We note here that in relaxing constraint (2) it is possible that $\sum_{j\in N} x_{ij} \geq 1$, which means that a customer $i \in M$ may be assigned to more than one facility.

The Lagrangian dual of (P) with respect to constraints (1) and (2) is Minimize $L(u, v)$, subject to $v \geq 0$. This dual can be solved via the following subgradient algorithm.

Subgradient algorithm

First, let Q be the incumbent solution set, i.e., the best set of locations for open facilities found so far, γ be the objective value of Q (i.e., γ is the best upper bound obtained so far for the primal problem (P)), and β be the best objective value found for the Lagrangian dual.

Step 1. Initialize by setting $\beta = -\infty$, $k = 1$; $v^1 = 0$ and $u_i^1 = \min\{d_{ij}: j \in Q\}$; $i \in M$.

Step 2. Find an optimal solution y_j^k, x_{ij}^k, $\forall i \in M, j \in N$ of the Lagrangian relaxation $L(u^k, v^k)$ by using Eqs. (8) and (9). Let

$$\theta_i = 1 - \sum_{j\in N} x_{ij} \ \forall i \in M,$$

and

$$\eta = b - \sum_{j\in N_1} c_j(1 - y_j) + \sum_{j\in N_2} c_j y_j.$$

If $\theta_i = 0$; $\forall i \in M$, $\eta \geq 0$ and $v^k = 0$, stop; y_j^k,x_{ij}^k solves (P).

Step 3. Update the value β if y_j^k, x_{ij}^k provides a better objective function value for the dual.

Step 4. Define a step size $\lambda_k = (\gamma - \beta)/(\|\theta\| t^k)$, where t^k is a factor related to iteration k. Update the Lagrangian multipliers by $u_i^{k+1} \leftarrow u_i^k + \theta_i\lambda_k$, $\forall i \in M$ and $v^{k+1} \leftarrow \max\{0, v^k + \eta\lambda_k\}$. Set $k \leftarrow k + 1$. If k is greater than the number of iterations permitted, stop; otherwise, go to step 2.

The following LR approximation scheme uses this subgradient algorithm to obtain an ε-optimal solution to problem (P):

LR approximation algorithm

Step 1. Solve the problem with the GI heuristic to obtain an initial feasible solution and an upper bound.

Step 2. Solve the problem via a branch-and-bound technique. The lower bound is obtained by solving the Lagrangian dual via the subgradient algorithm presented previously. Branch on the y_j with the most customers assigned to it. A node with lower bound no less than $\gamma/(1 + \varepsilon)$ will be pruned from the branch-and-bound tree, where γ is the best objective function value of (P) found so far.

The reason for branching on the facility with the most customers is because this facility stands a good chance to be in the optimal solution (in which case its y value will be set to 1). With this in mind, we hope to reduce the size of branch-and-bound tree by fixing such variables at an early stage.

Obviously the CPU time of the LR approximation will be larger than the CPU time of the GI heuristic since the GI procedure is a part of the LR approximation. However, the benefit of the LR approximation is that it always provides a solution that is no worse than the one provided by the GI heuristic and it guarantees that the solution is within 8% of the optimum.

5. Computational experiment

In this section, we test the performance of the heuristic algorithms. This testing is based upon the application domain of locating bank branches in a large-size township. We chose Amherst, New York, which has a population of about 100,000 people (it is the largest suburb of the city of Buffalo, New York) as the region of interest. By looking at the major banks in this town we found that the number of branches varied between 5 and 10. Also, by talking with officials in the town office it was determined that only commercially zoned land at major street intersections would be considered for location of a bank branch due to consideration of parking, accessibility, etc. In the description we highlighted the choice of the problem parameters with respect to these application domain.

All the three heuristics were compared to the exact solution for 270 problems. Test data are based on a network derived from the Town of Amherst, New York. It consists of 459 customer nodes and 84 potential facility sites. Since link information plays no role in all the three heuristics, the distance information between each (customer, facility) pair is enough. Heuristic algorithms were programmed in Microsoft Visual C++ 6.0. The optimal solutions were generated via CPLEX 6.0. All the programs were run on a Dell OptoPlex GX1MTDS450 PC with 128MB RAM and a 450 MHz CPU. Computation times are reported in seconds.

5.1. Parameters

The following is a list of the parameters considered in the numerical experiment:

- Customer nodes and weights
  We obtained the customer nodes and their weights from census block data from the 1991 US Census. Essentially, the weight of a node was the portion of the population of the town that resided in the area represented by that node in the 1991 census.
- Distance matrices
  Ten weighted distance-matrices were generated by perturbing the Euclidean distance between the customer nodes and the potential facility nodes. Let e_{ij} be the Euclidean distance between customer node i and potential facility node j. Then $d_{ij} = \alpha e_{ij}$, where the perturbation α was randomly generated from a uniform distribution over (1.1, 1.4) for each link. The range for perturbation stems from the fact that road distances between two randomly selected points in the Town of Amherst tend to be mainly within 10–40% of the Euclidean distance between the points (based on some experimentation we did).
- Opening/closing costs
  The opening cost was randomly generated from a uniform distribution over (200, 300). The closing cost was randomly generated from a uniform distribution over (50, 100).
  The closing cost is smaller than the opening cost because opening a new facility is typically more costly than closing an existing facility. We note that the scale of these costs is not relevant. However, the relative magnitude of opening and closing costs are of importance and were obtained by consulting with a local real-estate agent with reference to business properties.
- Number of existing facilities

The parameter q has three levels: {5,7,10}. The set of the existing facility sites was randomly picked from the 84 potential facility sites. The number of potential facility sites were chosen with a bank example in mind. Specifically, we identified commercially zoned land/properties at major street intersections in the town of Amherst. The values of q reflect the number of branches associated with the three largest banks in the Town of Amherst.

- Number of open facilities
  The parameter p has three levels {5,10,15}. The values of p reflects the fact that a bank may wish to expand its services from its current level.
- Budget
  The budget b is determined by q, p and a constant c. The constant c also has three levels { 1.5,1.8,2}. The value of b is given by

$$b = \begin{cases} 250c(p-q) if\ p > q, \\ 500c \quad\quad\quad if\ p = q, \\ 125c(q-p) if\ p < q. \end{cases}$$

  Since we have chosen opening costs to be higher than closing costs we allow for a larger budget when we are permitting a net increase in the number of facilities ($p > q$). For the case $p = q$, we permit a budget which allows for closing/opening of a reasonable number of facilities. The three values of c allow for a change in the overall budget while retaining its variation across p and q.
- Parameters for the TS approach
  The length of the tabu list is $L = 3$. The maximal number of non-improvement swaps allowed is $K = 5$. The values of $L = 3$ and $K = 5$ were chosen after initial experimentation with the TS method.
- Parameters for the LR approximation
  The parameter $\varepsilon = 0{:}02$. We chose this value since most decision makers would be satisfied with a solution that is within 2% of optimality.
  The number of iterations permitted in the subgradient algorithm is 300. This value was chosen after initial experimentation with the TS method.
  The step size λ^k is defined by $\lambda^k = (\gamma - \beta)/(\|\theta\| t^k)$, where γ is the best objective function value of (P) found so far, $\|\theta\| = (\sum_i \theta_i^2)^{1/2}$ is the norm of θ, and

$$t = \begin{cases} 50 \text{ if } k \leq 5, \\ 25 \text{ if } 5 < k \leq 20, \\ 15 \text{ if } 20 < k \leq 30, \\ 5 \text{ if } 30 < k \leq 40, \\ 1 \text{ otherwise.} \end{cases}$$

  As is standard practice in applying LR as a solution technique Beasley [30], the step length should be decreased significantly as the iterations progress. The values of t chosen reflect this desire.

Since for each distance matrix there are three parameters q, p and c, and each parameter has three levels, the number of combinations of the levels of these three parameters is $3 \times 3 \times 3 = 27$. For each combination of these parameters, we ran 10 problems corresponding to the 10 distance matrices. Hence, there were 270 problems in total. All three heuristics were used to solve these 270 problems.

We have also tried to revise the greedy phase of the GI heuristic by using a simulated annealing technique. For most of the 270 problems we tested, the revised algorithm generated worse solutions. Therefore, its results are not reported.

5.2. Computational results

For all the algorithms, the CPU time and objective function value are collected for each problem. The error percentage of the heuristics is computed as

$$\text{error} = \frac{\text{Heuristic objective value} - \text{optimal objective value}}{\text{Optimal objective value}} \times 100\%$$

where the optimal objective value was determined using the commercial optimization software CPlex 6.0.

Table 1
Objective value errors for the approximating heuristics

q	p	c	Average GI error (%)	Average TS error (%)	Average LR error (%)	Max GI error (%)	Max TS error (%)	Max LR error (%)
		1.5	0.67	0.14	0.24	3.44	1.38	1.54
	5	1.8	0.60	0.38	0.22	3.78	3.78	1.85
		2	1.14	0.56	0.08	4.53	2.68	0.47
		1.5	0.26	0.08	0.26	1.32	0.49	1.32
5	10	1.8	0.62	0.45	0.62	1.48	1.48	1.48
		2	0.73	0.59	0.10	3.11	0.72	1.73
		1.5	0.48	0.36	0.48	1.26	1.26	1.26
	15	1.8	0.45	0.37	0.45	1.52	1.52	1.52
		2	0.45	0.37	0.45	1.52	1.34	1.52
		1.5	0.00	0.00	0.00	0.00	0.00	0.00
	5	1.8	1.16	0.06	0.14	10.26	0.61	0.74
		2	0.25	0.09	0.25	1.80	0.92	1.80
		1.5	0.11	0.00	0.00	1.09	0.00	0.00
7	10	1.8	1.05	0.42	0.79	2.18	1.50	1.78
		2	0.34	0.21	0.34	1.14	1.10	1.14
		1.5	0.54	0.41	0.50	1.54	1.54	1.17
	15	1.8	0.49	0.34	0.49	1.49	0.82	1.49
		2	0.60	0.40	0.60	1.14	1.14	1.14
		1.5	0.00	0.00	0.00	0.00	0.00	0.00
	5	1.8	0.67	0.28	0.04	2.98	2.41	0.25
		2	0.62	0.31	0.29	2.98	2.41	0.70
		1.5	0.00	0.00	0.00	0.00	0.00	0.00
10	10	1.8	0.00	0.00	0.00	0.00	0.00	0.00
		2	0.00	0.00	0.00	0.00	0.00	0.00
		1.5	0.35	0.35	0.35	0.82	0.82	0.82
	15	1.8	0.52	0.37	0.52	1.44	1.29	1.44
		2	0.26	0.18	0.24	0.74	0.74	0.56
Average			0.46	0.25	0.27	1.91	1.11	0.95

The average error percentage and maximal error percentage for each group is listed in Table 1. The first three columns display the average (over the group of 10 problems with specified values of parameters q, p and c) of the error between the objective function value obtained by the three heuristics and the optimal objective value. The last three columns of the table are the maximum (over the 10 problems) of the error between the objective value obtained by the three heuristics and the optimal value.

Based on these results, we can conclude that the performance of all three heuristics is very good. The average error of the GI heuristic across all parameter combinations is only 0.46%. (Only one problem among the 270 had an error larger than 10%.) The average error of the TS approach is just 0.25%. (Only one problem among the 270 had an error larger than 3%.) Similarly, the average error of the LR approximation is 0.27%.

Table 2
The average computation time for the heuristics and CPLEX

q	p	c	GI	TS	LR	CPLEX
		1.5	1.1	1.6	15.9	1482.3
	5	1.8	1.2	1.6	68.2	2171.2
		2	1.1	1.7	37.0	1487.6
		1.5	3.6	6.1	5.6	811.2
5	10	1.8	4.5	7.5	10.9	644.9
		2	5.5	8.6	20.7	596.6
		1.5	10.7	19.9	16.2	297.0
	15	1.8	11.9	21.3	17.1	231.1
		2	11.5	21.6	17.1	205.8
		1.5	0.6	0.6	103.4	2013.6
	5	1.8	0.7	0.7	71.8	1343.6
		2	0.8	1	10.5	768.5
		1.5	2.4	3.3	10.2	1268.3
7	10	1.8	2.6	3.6	36.2	1202.6
		2	3.3	4.7	9.0	1208.1
		1.5	9.3	15.7	25.0	601.0
	15	1.8	13.2	20.2	18.8	364.9
		2	14.7	24.9	18.2	264.9
		1.5	0.7	1.2	93.0	2099.9
	5	1.8	1.2	1.3	13.5	1482.9
		2	1.1	1.6	60.1	1820.1
		1.5	1.2	1.5	46.6	422.2
10	10	1.8	1.4	1.7	97.4	1096.4
		2	1.4	1.8	122.0	1872.1
		1.5	5.0	7.8	7.3	673.1
	15	1.8	7.3	11.9	10.0	830.8
		2	7.6	11.7	10.4	653.7
Average			4.7	7.6	36.0	1033.9

It is important to note that the GI heuristic and TS approach provide optimal solutions in most of the problems when $p \leq q$. This shows that the algorithms are well suited for the situation when a business is consolidating rather than expanding its locations.

However, the GI heuristic and the TS approach make no guarantees on the quality of the solution obtained, whereas the LR approximation solutions are guaranteed to be within 2% of the optimal value. Among the 270 problems, the solutions of 12 problems generated by the GI heuristic and the solutions of five problems generated by the TS approach were beyond 2% of the optimal values.

The results in Table 2 are the average time spent by the three heuristics as compared to the time spent by CPLEX 6.0 for a group with specified values of parameters q, p and c. On average (i) the TS approach takes 60% more time than the GI heuristic; (ii) the LR approximation takes 373% more time than the TS approach; and (iii) the time spent on CPLEX 6.0 is almost 28 times of that by LR approximation.

For the GI heuristic and the TS approach, the CPU time generally

- increases in the overall budget c; this is because when the budget is large, the interchange phase is less restricted;
- increases in p; the reason for this is that when p is large, the number of possible swaps in each iteration of the interchange phase is large;
- decreases in the value of $p - q$.

The above three results do not apply to the LR approximation and CPLEX. The reason may be due to the fact that both of these methods use a branch-and-bound scheme, whose convergence depends on the lower bounds it finds. Apparently parameters p; q and c do not have a strong influence on the quality of the lower bounds obtained.

5.3. CPLEX approximation

We also used CPLEX as an approximation method by terminating it when the solution at hand was within 2% (or 3%) of optimality. As one might expect, this CPLEX approximation methods generates solutions that are sometimes better and sometimes worse than the GI heuristic and the LR approximation. (The TS approach is similar to the GI heuristic.) However, it always takes much more time than the other heuristics. Since it generally takes more time to obtain a better solution, we omitted those cases where CPLEX provided a better solution than the heuristics. This was done in order to make a fair comparison to the CPLEX approximation. If this were not done, then the CPLEX "heuristic" would be unfairly penalized by instances of excessively large computation time — a price that was paid solely because the other heuristics were unable to come within 2% (or 3%) of optimality.

Tables 3 and 4 compare the CPU times for those problems that the CPLEX approximation provided worse solutions than the heuristics. From these tables, we can readily see that the time that CPLEX approximation method takes is about 183 times that of the GI heuristic and 25 times that of the LR approximation. In these instances, the GI heuristic and the LR approximation generate better solutions with much less time than the CPLEX approximation, and therefore are capable of handling larger problems. Hence, the development of these heuristics appears to be worthwhile.

5.4. Large-scale problems

We ran the three heuristics on the same PC to solve four randomly generated large-scale problems. These problems could not be solved via CPLEX 6.0 on this PC due to the lack of memory. The description of the problems is in Table 5.

The computational results for both solution time and objective value are presented in Table 6. The first two rows of the table show the performance of the GI heuristic and the TS approach, respectively. The remainder of the table shows the performance of the LR approximation for decreasing values of error tolerance 8.

Based on these results, we can conclude that the heuristics are still efficient for large-scale problems. The GI heuristic generated good approximating solutions and provides good upper bounds for the LR approximation. On the four problems, the solutions achieved by the GI heuristic were within 0.4%, 0.1%, 1% and 3% of optimality, respectively. Although slightly slower, the TS approach provided better solutions than the GI heuristic for three of the four problems. On the four problems, the solutions achieved by the TS approach were guaranteed to be within 0.4%, 0.1%, 0.1% and 0.6% of optimality, respectively.

In fact, on problem four the TS approach achieved the best solution found by the time-limited LR approximation. Hence, the 0.6% guarantee is a conservative one. The computation times for the LR approximation as a function of 8 are plotted in Fig. 1. The CPU time increases dramatically when 8 goes to 0 with no significant change in the objective function value.

Table 3
Comparison of the average computation time for the GI heuristic and the CPLEX approximation (when worse)

q	p	c	GI	CPLEX	CPLEX/GI
		1.5	1.25	1396.63	1117.30
	5	1.8	1.17	1862.78	1592.12
		2	1.33	1245.67	936.59
		1.5	3.64	795.43	218.52
5	10	1.8	4.70	476.60	101.40
		2	6.50	656.67	101.03
		1.5	10.83	212.83	19.65
	15	1.8	12.50	245.33	19.63
		2	12.30	222.40	18.08
		1.5	0.60	1878.80	3131.33
	5	1.8	0.61	1291.56	2117.31
		2	0.75	738.75	985.00
		1.5	2.50	993.44	397.38
7	10	1.8	2.75	1068.50	388.55
		2	3.30	954.00	289.09
		1.5	9.83	549.83	9.00
	15	1.8	12.17	428.00	35.17
		2	15.17	247.33	16.30
		1.5	0.70	1778.80	2541.14
	5	1.8	1.30	722.60	555.85
		2	1.13	1428.25	1263.94
		1.5	1.20	423.40	352.83
10	10	1.8	1.40	1000.00	714.29
		2	1.40	1587.60	1134.00
		1.5	4.93	430.43	87.31
	15	1.8	7.39	524.33	70.95
		2	7.64	463.14	60.62
Average			4.78	874.93	183.14

6. Operating costs

In this section, we extend our model to the situation where operating costs are also considered over a fixed time horizon. Generally, the operating cost of an open facility site j is given by $f_j + g_j x_j$,

Table 4
Comparison of the average computation time for the LR approximation and the CPLEX approximation (when worse)

q	p	c	LR	CPLEX	CPLEX/LR
		1.5	19.00	1396.63	73.51
	5	1.8	62.28	1862.78	29.91
		2	32.50	1328.50	40.88
		1.5	6.17	795.43	128.92
5	10	1.8	7.30	476.60	65.29
		2	8.50	656.67	77.26
		1.5	14.83	212.83	14.35
	15	1.8	16.33	245.33	15.02
		2	16.50	222.40	13.48
		1.5	103.90	1878.80	18.08
	5	1.8	72.30	1266.50	17.52
		2	8.13	738.75	90.87
		1.5	10.70	972.40	90.88
7	10	1.8	34.30	927.40	27.04
		2	9.50	954.00	100.42
		1.5	13.83	549.83	9.00
	15	1.8	16.17	428.00	26.47
		2	19.17	247.33	12.90
		1.5	93.50	1778.80	19.02
	5	1.8	11.38	723.38	63.57
		2	60.60	1428.25	23.57
		1.5	47.10	423.40	8.99
10	10	1.8	97.90	1000.00	10.21
		2	122.50	1587.60	12.96
		1.5	7.79	430.43	55.25
	15	1.8	10.61	524.33	49.42
		2	10.93	463.14	42.37
Average			34.58	871.09	25.19

Table 5
Parameters of the four large-scale problems

Problem	m	n	q	p	b
1	1500	100	15	19	2000
2	1500	100	12	15	2500
3	1500	100	11	14	1500
4	1000	100	7	9	1000

where

f_j is the fixed operating cost for facility j over the time horizon;
g_j is the operating cost at facility j per unit customer demand;
x_j is the customer demand at facility j.

Table 6
Computational results for the large-scale problems

		Solution time				Objective value			
		1	2	3	4	1	2	3	4
GI		35	34	26	4	141,731	158,966	169,706	146,102
TS		56	51	37	9	141,603	158,966	169,618	145,203
LR ε (%)	3.0	48	51	38	13	141,731	158,966	169,706	146,102
	2.0	48	51	38	168	141,731	158,966	169,706	145,957
	1.8	48	51	38	336	141,731	158,966	169,706	145,957
	1.6	48	51	38	662	141,731	158,966	169,706	145,957
	1.4	48	51	38	590	141,731	158,966	169,706	145,435
	1.2	48	51	100	844	141,731	158,966	169,706	145,435
	1.0	48	51	164	1935	141,731	158,966	169,706	145,417
	0.8	48	51	210	2358	141,731	158,966	169,065	145,203
	0.6	48	51	224	4560	141,731	158,966	169,065	145,203
	0.4	48	83	1444	> 1 h	141,731	158,966	169,065	
	0.2	426	1020	2714		141,567	158,966	168,886	
	0.1	1573	4720	5338		141,567	158,966	168,886	
	0.0	> 1 h	> 1 h	> 1 h					

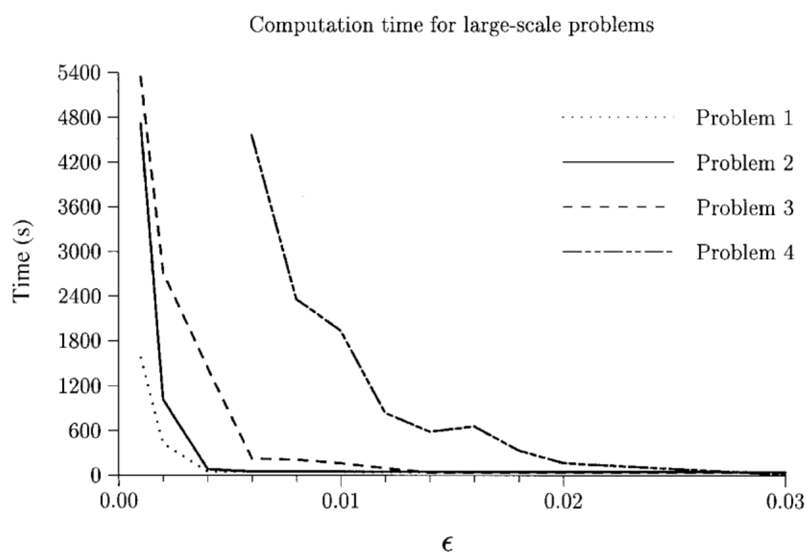

Fig. 1. Computation time of the LR approximation for large-scale problems.

Let h be the customer cost per unit distance travelled, and suppose that the customer demands w_i's represent demand over the planning time horizon. Then the objective function of Problem (P) becomes

$$\text{Minimize} \sum_{i\in M}\sum_{j\in N} hw_i d_{ij} x_{ij} + \sum_{j\in N}(f_j + g_j x_j)y_j.$$

In the case that the g_j are different for each facility j, the problem becomes non-linear. A solution approach to this problem would be to adopt a branch-and-bound method in which we branch on the variable y_j. For a fixed set of y_j's we can solve the problem quite easily. When the set of y_j's is partially fixed we could solve a linear relaxation of the problem using a non-linear solver and then branch on either $y_j = 0$ or 1 for a specified value of j. The development of such a solution strategy is beyond the scope of the current paper and is suggested as a future research topic.

We now consider the easier case, where g_j is the same for each facility, i.e., $g_j = g$, $\forall j$, the total variable cost would be $g \sum_{i \in N} w_i$, which is a constant no matter where the facilities locate. Thus, these costs can be ignored so that we just need to consider the fixed operating costs. Since the objective now considers operating costs for the service provider in addition to the travelling costs of the customers, the number of facilities should now be a decision variable. Hence, we drop constraint (3) on the number of facilities to open. In what follows, we discuss the extension of the GI heuristic and the LR approximation methods to this problem (the TS approach remains the same).

6.1. GI heuristic

This algorithm consists of two phases: greedy adding/dropping and swapping. In the greedy adding/dropping phase, we either open or close facilities until the objective value cannot be further improved by only opening or closing one facility. Refer to Section 4.1 for notation.

If a new facility is added, the total weighted travel distance will either remain the same or decrease, but the operating cost will increase. The total change of the objective function after adding a facility at node j can be evaluated by calculating the quantity

$$\phi_j = f_j - h \sum_{i \in M} w_i \, max\left(0, \delta_i - d_{ij}\right).$$

If an open facility is closed, the total weighted travel distance will either be unchanged or increase, but the operating cost will decrease. The total change of the objective function after closing a facility at node j is obtained by the expression

$$\phi_j = h \sum_{i \in M} w_i \min\{(d_{ik} - \delta_i): k \in Q\{j\}\} - f_j.$$

The criterion for selecting the facility to open/close at each step is to select the one with the smallest value of ϕ_j/c_j from those nodes satisfying $\phi_j < 0$ and $C(Q \cup \{ j\}) < b$ if adding a facility ($C(Q \setminus \{ j\}) < b$ if closing a facility).

Phase two is an interchange phase that performs a swap between a pair of facility nodes. The pair should consist of a node $j \in N$ with an open facility and a node $k \in N$ without an open facility. If there exist feasible swaps that improve the objective function value, the one with the best objective function value is selected. Then we apply the greedy adding/dropping procedure to the new facility set. The interchange procedure continues until there are no feasible swaps or no feasible swaps improve the objective function value.

6.2. LR approximation

The branch-and-bound scheme described in Section 4.3 can still be applied to the new problem. Everything is the same, except the Lagrangian relaxation solution. Consider the LR for the new problem with respect to constraints (1) and (2) for a given (u, v):

$$L'(u,v) = \min_{x,y} \sum_{i \in M} \sum_{j \in N} h w_i d_{ij} x_{ij} + \sum_{j \in N} f_j y_j + \sum_{i \in M} u_i \left(1 - \sum_{j \in N} x_{ij}\right)$$

$$+ v \left[\sum_{j \in N_1} c_j (1 - y_j) + \sum_{j \in N_2} c_j y_j - b\right]$$

subject to constraints (4)–(6).

The objective function in the relaxation $L'(u, v)$ can be rewritten as

$$\sum_{i\in M}\sum_{j\in N} h(w_i d_{ij} - u_i)x_{ij} + \sum_{j\in N_1}(f_j - vc_j)y_j + \sum_{j\in N2}(f_j + vc_j)y_j + \sum_{i\in M} u_i + v\sum_{j\in N_1} c_j - vb. \qquad (10)$$

Let

$$\rho_j(u,v) = \begin{cases} h\sum_{i\in M}(w_i d_{ij} - u_i)^- + f_j - vc_j & \text{if } j\in N_1, \\ h\sum_{i\in M}(w_i d_{ij} - u_i)^- + f_j + vc_j & \text{if } j\in N_2. \end{cases}$$

Since the last three terms in (10) are constant, it follows by inspection that the optimal solution to the LR for a given u and v is

$$y_j = \begin{cases} 1 & \text{if } p_j < 0, \\ 0 & \text{otherwise;} \end{cases}$$

$$x_{ij} = \begin{cases} y_j & \text{if } w_i d_{ij} - u_i < 0, \\ 0 & \text{otherwise.} \end{cases}$$

7. Summary, conclusions and future research

In this paper we have introduced and studied a budget constrained facility relocation problem where the facilities may be opened and closed simultaneously. Following the formulation and a note of its NP-hardness, three heuristics are developed to solve this problem. Numerical experiments have been run to examine the quality of the heuristics, and the results are reported. The model has also been extended to the situation where operating costs are considered.

Based on the results of numerical experiments we conclude that the performance of the three heuristics is very good. We also show the tradeoff between time and solution quality. The GI algorithm is the fastest, followed by the TS approach and then the LR approximation. However, the solutions generated by the TS approach are better than the GI heuristic. Further, the LR approximation can guarantee the precision of solution while the other two heuristics cannot. The heuristics also perform well for large problems, which demonstrates the superiority of these specialized solution algorithms over a generic IP solver such as CPLEX.

In terms of future research, we note that we have assumed that each facility has an unlimited capacity to service demands, which is clearly not suitable for the situation where capacity plays an important role. Hence, future research should examine versions of this problem and develop some efficient algorithms for them. A related avenue of future research would be to examine probabilistic versions of this problem when queue capacity is of concern. Another topic of interest might be studying the situation where closing costs are negative; this may be relevant under certain situations since the savings due to closing of a facility are accrued annually. A final topic of future research is the case of operating costs when each facility would have a different cost per unit customer demand. As noted in Section 6, a branch-and-bound approach may be viable for this situation.

Acknowledgements

The authors would like to thank two anonymous referees for their helpful comments on earlier versions of this paper.

References:

[1] Hakimi SL. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 1964;12:450–9.
[2] Levy J. An extended theorem for location on a network. Operational Research Quarterly 1967;18:433–42.
[3] Handler GY, Mirchandani P. Location on networks, theory and algorithms. Cambridge, MA: MIT Press, 1979.
[4] Kariv O, Hakimi SL. An algorithmic approach to network location problems, part 2: the p-medians. SIAM Journal of Applied Mathematics 1979;37:539–60.
[5] Cornuejols G, Fisher ML, Nemhauser GL. Location of bank accounts to optimize float: an analytical study of exact and approximate algorithms. Management Science 1977;23:789–810.
[6] Maranzana FE. On the location of supply points to minimize transport costs. Operational Research Quarterly 1964;15:261–71.
[7] Teitz BC, Bart P. Heuristic method for estimating the generalized vertex median of a weighted graph. Operations Research 1968;16:955–61.
[8] Jarvinen P, Fajala J, Sinervo H. A branch-and-bound algorithm for seeking the p-median. Operations Research 1972;20:173–82.
[9] Captivo ME. Fast primal and dual heuristic for the *p*-median location problem. European Journal of Operational Research 1991;52:65–74.
[10] Maniezzo V, Mingozzi A, Baldacci R. Binomic approach to the capacitated *p*-median problem. Journal of Heuristics 1998;4:263–80.
[11] Rosing KE, ReVelle CS, Schilling DA. Gamma heuristic for the *p*-median problem. European Journal of Operational Research 1999;117:552–632.
[12] Cornuejols G, Nemhauser GL, Wolsey LA. The uncapacitated facility location problem. In: Mirchandani PB, Francis RL, editors. Discrete location theory. New York: Wiley, 1990. p. 119–71.
[13] Kuehn AA, Hamburger MJ. A heuristic program for locating warehouse. Management Science 1963;9:643–66.
[14] Erlenkotter D. A dual-based procedure for uncapacitated facility location. Operations Research 19T8;26:992–1009.
[15] Tcha DW, Ro HB, Yoo CB. A dual-based add heuristic for uncapacitated facility location. Journal of the Operational Research Society 1988;39:873–8.
[16] Balinski ML. Integer programming: methods, uses, computation. Management Science 1965;12:253–313.
[17] Denshan P, Rushton G. Strategies for solving large location-allocation problem by heuristics. Environment and Planning 1992;A 24:289–304.
[18] Jones P, Lowe TJ, Muller G, Xu N. Specially structured uncapacitated facility location problems. Operations Research 1995;43:661–9.
[19] Berman O, Simchi-Levi D. Conditional location problems on networks. Transportation Science 1990;24:77–8.
[20] Drezner Z. On the conditional *p*-median problem. Computers and Operations Research 1995;22:84–91.
[21] Chhajed D, Lowe TJ. An O(nm) algorithm for a special case of the multi-median location problem on a tree. European Journal of Operational Research 1992;63:222–30.
[22] Leorch AG, Boland N, Johnson EL, Nemhauser GL. Finding an optimal stationing policy for the US army in Europe after the force drawdown. Military Operations Research 1996;2:39–51.
[23] Dell R. Optimizing army base realignment and closure. Interfaces 1998;28:1–18.
[24] Associated Press. Wells funds ATM access study. The Daily Review, Stanford, 1998.
[25] Min H, Melachrinoudis E. The relocation of a hybrid manufacturing/distribution facility from supply chain perspectives: a case study, Omega. The International Journal of Management Science 1999;27:75–85.

[26] Melachrinoudis E, Min H. The dynamic relocation and phase-out of a hybrid, twoechelon plant/warehousing facility: a multiple objective approach. European Journal of Operational Research 2000;33:1–15.
[27] Wang Q. Discrete facility location design with stochastic customer demand and immobile servers. Unpublished Ph.D. Dissertation, Department of Industrial Engineering, University at Buffalo (SUNY), 2000.
[28] Glover F, Laguna M. Tabu search. In: Reeves CR, editor. Modern heuristic techniques for combinatorial problems. New York: Wiley, 1993. p. 70–150.
[29] Narula SC, Ogbu VI, Samuelsson HM. An algorithm for the p-median problem. Operations Research 1977;25: 709–12.
[30] Beasley JE. Lagrangian Relaxation. In: Reeves CR, editor. Modern heuristic techniques for combinatorial problems. New York: Wiley, 1993. p. 243–303.

Qian Wang received her M.S. degree in Operations Research from Institute of System Science, Chinese Academy of Sciences, and her Ph.D. degree in Operations Research from SUNY at Buffalo. Her current interests are in the areas of supply chain management, scheduling, queueing, network design and large-scale optimization.

Rajan Batta is Professor and Chair of the Department of Industrial Engineering at the State University of New York at Buffalo, USA. His interests are in Applied Operations Research, particularly in the effective modelling of probabilistic events. He has a B.Tech. degree from the Indian Institute of Technology, New Delhi, and a Ph.D. in Operations Research from the Massachusetts Institute of Technology.

Joy Bhadury is a Professor of Quantitative Business Management at California State University—Hayward. He has taught previously at SUNY, Buffalo, University of New Brunswick in Canada. His research has been published in European Journal of Operational Research, INFORMS Journal of Computing, Computers and Operations Research, Journal of Regional Science, Social Choice and Welfare, Location Science OR Spektrum. His research and teaching specializations are in logistics management and management science.

Christopher M. Rump is an Assistant Professor in the Department of Industrial Engineering at the University at Buffalo, State University of New York. He received a B.S. in Applied Mathematical Sciences from Texas A& M University in 1989 and a Ph.D. in Operations Research from The University of North Carolina at Chapel Hill in 1995. His research interests are in applied probability modelling with applications to a mix of theoretical models for queueing, inventory, location, games and decisions. His research and educational articles have been published in a variety of journals including Computers and Operations Research, Management Science, IIE Transactions, and the INFORMS Transactions on Education.