Multi-Period Facility Location Problem with an Uncertain Number of Servers

Amit Vatsa

W.P. No. 2014-02-06
February 2014

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD – 380015
INDIA

Multi-Period Facility Location Problem with an Uncertain Number of Servers

Amit Vatsa

Abstract

We study the problem of allocating doctors to primary health centers. We model the problem as a multi-period uncapacitated facility location problem under uncertainty. The problem is unconventional in that the uncertainty is in the number and period of availability of doctors. We use a minmax regret approach to solve the problem. We present solution techniques using local search and tabu search and compare our solutions with optimal solutions obtained using commercial solvers. We see that one of our tabu search algorithms is faster and yields optimal solutions in the problems we tested on.

Keywords: Facility location, Heuristics, Tabu search

1 Introduction

This study has been motivated by the primary healthcare sector in the developing countries. Providing basic health care to its citizens is an important function of any government. Almost all the developing countries have rural populations that are larger than urban populations. Despite this, healthcare facilities are mostly concentrated in the urban regions. Policy makers try to minimize this disparity and make basic health amenities available to everyone in the rural areas as well.

In the Alma-Ata declaration of 1978 by the members of World Health Organization (WHO), a need for primary health care was reiterated. It was suggested in the declaration that there should be one Primary Health Center (PHC), which is essentially a single doctor clinic, for a population of 30,000 in plains and 20,000 for hilly regions. Many developing countries have tried to achieve this goal. However, due to many reasons this target could not be fulfilled (Walley et al., 2008; Rohde et al., 2008). Rapid growth in populations in most of these countries require that the targets should be revisited. With the economic development in many of these countries and the resulting rise in income, the urban-rural disparity will be huge if basic amenities are not made available at the local level in rural areas. Locating these PHCs so as to meet all the local demand is a major challenge for the government. The government tries to establish a network of PHCs so that basic health care facilities are available to the largest possible rural population.In most of the developing countries, availability of doctors in rural areas is a constraint, even if the physical infrastructure exists (Lawn et al., 2008). When doctors become available it needs to be decided to which among those PHCs without a doctor, should the doctor be assigned.

We define the uncapacitated PHC location problem(UPHCLP) of determining the sequence in which the PHCs should be assigned with a doctor. The numbers of doctors who become available in each period of the planning horizon is uncertain. Our solution will minimize the maximum regret from an *ex post* optimal population coverage, i.e., the population coverage which could have been acheived had the numbers of doctors who will join in each period of the planning horizon been known *a proiri.*

This paper attempts to model the UPHCLP and develop a solution methodology to determine the sequence of opening facilities, i.e., assigning doctors to the PHCs. The remainder of this paper is organized as follows. We first provide the overview of related existing literature in section 2. In section 3 we provide a formulation of the problem. In Section 4 we give some dominance rules to fasten the heuristics and discuss the local search and three tabu search based heuristics. In section 5 we report our computational experience with these heuristics and compare the solution with the solution given by CPLEX 12.4. In the last section we provide a summary of the current work and present some future research directions.

2 Related work

The general facility location problem involves two decisions, a location decision to decide where a facility should be set up, and an allocation decision to decide which customers will be served by a particular facility. Opening of new facilities involve time and capital investment, and it is one of the most important decisions for any institution. Hence, facility location models have been extensively used by the World Bank and various government projects. Some examples of the areas where they find application are location of schools (Antunes & Peeters, 2000, 2001); ambulance deployment (Brotcorne et al., 2003) and establishment of the network of health care facilities (Ghaderi & Jabalameli, 2012).

Facility location problems have been widely addressed in the literature (See, e.g., Drezner & Hamacher, 2001; Farahani & Hekmatfar, 2009). There are two predominant objective function variants which have been widely studied: minisum and minimax (Hale & Moberg, 2003). These are also known as median and center problems respectively. Other than these two objective functions, set covering and maximal covering objectives have also been studied for more than three decades.

Depending on number of periods in the planning horizon in which location and allocation decisions are to be made, facility location problems can be categorized as single-period or multi-period. Most of the early work was done for the static or single period case. Multi-period facility location problems are important because of two reasons. First, customer demands, transportation/assignment costs and other parameters change over time. Secondly, relocation of facilities involve capital expenditure. In absence of the first characteristic, a single period model can be used to solve the problem and, in the absence of second criteria a series of disconnected static formulations can be used (Erlenkotter, 1981). In presence of budget constraints all facilities cannot be opened at once and thus the multi-period problem becomes important.

Multi-period facility location problems have been widely studied after the initial works by Warszawski (1973); Erlenkotter (1981); Van Roy & Erlenkotter (1982). In the single period literature, p-median problem, UFLP, CFLP, p-center problem, set covering problem, and maximal covering location problem (MCLP) have been widely studied (See, e.g., Drezner & Hamacher, 2001; Farahani & Hekmatfar, 2009).

Work on the dynamic counterpart of the minisum problems like the p-median, UFLP, CFLP started in 1970s and 1980s. Problems under various constraints were looked at. Some example of these variants are constraints on location and relocation of facilities (Wesolowsky & Truscott, 1975; Melo et al., 2006), reopening cost different than first time opening cost (Dias et al., 2006, 2007, 2008), and non-zero closing cost (Wesolowsky & Truscott, 1975; Saldanha da Gama & Captivo, 1998; Canel et al., 2001).

There are many papers in the facility location literature which consider the uncertainty in decision making environment. Most of the problems which have been studied in the literature have uncertainty or incomplete information of the demand at the nodes (Killmer et al., 2001; Averbakh, 2003; Albareda-Sambola et al., 2011). Current et al. (1998) used the minimax regret criteria to decide where to set up the initial set of facilities when the total number of facilities to be set up in the future is uncertain. The procedure was presented in the context of p-median problem. Berman & Wang (2011) modeled the MCLP with partial coverages. In their problem demand at the nodes are random variables whose probability distributions are unknown. They used the information on the range of these variables to find the minimax regret location that minimizes the worst-case coverage loss. We now discuss the single period and multi-period version of the maximal covering location problem.

2.1 Maximal covering location problem

The MCLP was introduced by Church & ReVelle (1974). In the MCLP we are given a set I of demand nodes, a set J of candidate locations for facilities, a matrix $C = [c_{ij}]$ of distance between each $i \in I$ and each $j \in J$, a vector (d_i) of demand at each demand node, and a covering distance. Each facility has an infinite capacity. We are required to locate p facilities so that the maximum demand at the demand nodes can be met by the facilities. The variables are: $y_j = 1$ if a facility is opened at location j, 0 otherwise; $x_i = 1$ if demand node i is within a covering distance of some opened facility, 0 otherwise. The problem can be formulated as follows:

$$\text{MCLP: maximize } \sum_i d_i x_i \tag{1}$$

subject to

$$\sum_j y_j = p \tag{2}$$

$$x_i \leq \sum_j a_{ij} y_j \qquad \forall i \tag{3}$$

$$y_j \in \{0, 1\} \qquad \forall j \tag{4}$$

$$x_i = [0, 1] \qquad \forall i \in I \tag{5}$$

Here $a_{ij} = 1$ if demand node i is within covering distance from facility j, 0 otherwise. The objective in the above formulation maximize the total demand covered. Constraint (2) limits the number of facility to be opened. Constraint (3) ensures that any demand node will be covered only if a facility within a covering distance from that demand node is opened. In the above formulation x_i need not be declared an integer. As y_j are integers and x_i are constrained only by the first constraint in the formulation, x_i will be integers too.

Church & ReVelle (1976) discussed the links between the MCLP and the p-median problem. If the distance c_{ij} is in binary form such that $c_{ij} = 1$ if the actual distance is within the covering distance, else $c_{ij} = 0$, then the p-median problem reduces to MCLP. Thus, the MCLP can also be solved by any of the solution methods for the p-median problem. However, it is not necessary that any such method will have the same efficiency in the MCLP as well.

Chung (1986) discuss various applications of the MCLP. Daskin (2000) used MCLP to solve the p-center problem to optimality. An MCLP is solved with different covering distances. When all the demand is met at some least possible covering distance, the p-center problem is solved to optimality. Schilling et al. (1993) reviewed papers on covering problems. Most of the studies focus on improving reliability of the system by providing multiple coverage with applications in emergency services like ambulance, fire stations and blood banks.

Berman et al. (2003); Karasakal & Karasakal (2004); Berman et al. (2010) consider a generalization of the maximal cover location problem. Their model allows for partial coverage of the demand nodes, and the degree of coverage is a non-increasing function of the distance to the nearest facility. Linear coverage decay function and step-function have been considered by them to account for the partial coverage in the objective function.

2.2 Multi-period maximal covering location problem

There has been limited work on the multi-period covering problems. Gunawardane (1982) first introduced the maximal coverage location problem in the dynamic scenario. In the problem setup, location of the facilities and the possible relocation within the planning horizon was considered. Small problems of size 10 to 30 centers and 5 periods were solved by the standard solvers. Most of the times the LP relaxation yielded integer solutions. The problem is an extension of MCLP, and a superscript t has been used in the notation of the parameters to indicate the values at time period t. The variables in this formulation are: $y_j^t = 1$ if a facility is open at location j at time period t, 0 otherwise; $x_i^t = 1$ if demand node i is within a covering distance of some opened facility at time period t, 0 otherwise. The problem can be formulated as follows:

$$\text{DMCLP: maximize } \sum_t \sum_i d_i^t x_i^t \tag{6}$$

subject to

$$\begin{aligned} x_i^t &\leq \sum_j a_{ij} y_j^t && \forall i, t && (7)\\ y_j^t &\geq y_j^{t-1} && \forall j, t > 1 && (8)\\ \sum_j (y_j^t - y_j^{t-1}) &= p^t && \forall t && (9)\\ y_j^t &\in \{0,1\} && \forall j, t && (10)\\ x_i^t &= [0,1] && \forall i, t && (11) \end{aligned}$$

In the above formulation x_i^t's need not be declared integer. As y_j^t are integers and x_i^t are constrained only by the first constraint in the formulation, x_i^t will be integers too. The objective in the above formulation maximizes the total demand covered in the planning horizon. Constraint (7) ensures that any demand node will be covered only if a facility within a covering distance from that demand node is open. Constraint (8) guarantees that facility once opened will not be closed in the planning horizon. Constraint (9) limits the number of facilities to be opened.

Zarandi et al. (2013) was the only study dealing with the multi-period MCLP and used simulated annealing to solve problems of size 2500 demand nodes and 200 potential candidate locations. A lot of work has been done on these minisum problems and there are many efficient algorithms to solve even the large instances of such problems. However, the problems with minimax structure like the p-center problem are relatively difficult to solve as they are generally solved by repeatedly solving NP hard set covering problems.

Till now no work has been done which takes into account the uncertainty in number of servers, i.e. in which the numbers of facilities that will be opened in any period are not known beforehand are not known beforehand. In many practical problems this uncertainty exists. For example in the UPHCLP , physical infrastructure for the PHC exist at many locations but due to unavailability of doctors, complete service cannot be provided by these PHCs. Those PHCs where doctors are not present can only provide very limited healthcare facilities and hence its objectives are not met. Number of doctors joining rural healthcare in any district is not known *a priori.* When a doctor joins, they are allocated to an unmanned PHCs. The sequence of opening new facilities, i.e. allocating a doctor to a site where physical infrastructure already exists, is pre-decided for a planning horizon. The above mentioned uncertainty may lead to such a facility opening sequence being considered, which is far from the optimal *ex post* decision, i.e. the optimal decision had the number and timing of doctors joining been known *a priori.*

In this work we are given a set of demand nodes and a set of candidate facility locations. Each facility has infinite capacity but can only meet demand which fall within a covering distance. The number of facilities to be opened in various periods of planning horizon is uncertain and gives rise to different scenarios. The objective of the problem is to determine the sequence of opening facilities (staffing PHCs with doctors) which creates the least worst case deviation from the optimal over all scenarios. This solution thus minimizes the maximum regret across scenarios.

3 Problem formulation

When the future scenario i.e the number of facilities which will be opened in all periods of the planning horizon is known, DMCLP gives the location of the facilities. However, when there is uncertainty in the numbers of facilities that can be opened in each period of the planning horizon, the objective can be to minimize the worst case regret from the decision. In this problem the regret associated with any solution for a given scenario is the difference of the *ex post* optimal coverage for that scenario and the coverage obtained with the solution.

We model the problem as follows. There is a set of demand nodes $I = \{1, 2, .., m\}$ and a set of facilities $J = \{1, 2, .., n\}$ which are to be opened. The first period demand at each demand node i is given by d_i^1. The demand at a node i changes in each period of the planning horizon with a growth rate g_i, which is assumed to remain constant over the planning horizon. Hence, the demand at a node i in period t of the planning horizon is given by $d_i^t = d_i^1(g_i)^{t-1}$.

Let $s \in S$ be the index which represents the future scenario in which new servers will be available at each period of the planning horizon. Thus, a scenario $(a_1, a_2, ..., a_T)$ indicates that in the first period, a_1 new servers will be available (a_1 facilities can be opened), a_2 new servers in the second period, likewise and in the last period a_T new servers will be available. With n candidate facilities to be opened in $|T|$ periods, number of scenarios of getting new servers over the planning horizon is $\binom{n+|T|-1}{n} = \frac{(n+|T|-1)!}{n!(|T|-1)!}$. For $|T| = 5$, total number of scenarios will be 126, 1001 and 3876 when n is 5, 10 and 15 respectively.

A solution in this problem is the facility opening permutation, and we introduce a variable $R_{j'j}$ to represent that. $R_{j'j} = 1$ if facility j has a rank j' of opening in a facility opening permutation, 0 otherwise. The variables of the DMCLP have been retained with an addition of subscript s to represent the values with the scenario $s \in S$. $y_{js}^t = 1$ if a facility is open at location j at time period t when the facility opening scenario is s, 0 otherwise; and, $x_{is}^t = 1$ if demand node i is within a covering distance of some opened facility at the time period t and scenario s; 0 otherwise. To establish a relation between the variables $R_{j'j}$ and y_{js}^t we can argue that a facility at j is open if and only if rank of facility j is less than the total number of facilities opened till that instant. Mathematically,

$$y_{js}^t = 1 \iff \sum_{j'} j' R_{j'j} \leq \sum_{t' \leq t} p_s^{t'}$$

Which translates into these two inequalities:

$$\sum_{j'} j' R_{j'j} \leq \sum_{t' \leq t} p_s^{t'} + n(1 - y_{js}^t) \tag{12}$$

$$\sum_{j'} j' R_{j'j} \geq \sum_{t' \leq t} p_s^{t'} + 1 - n y_{js}^t \tag{13}$$

Here p_s^t is the number of new servers which become available at period t when the scenario is s. Let Π be the set of all possible facility opening permutation and Π_k be one of the facility opening permutations. Demand coverage with Π_k in scenario $s \in S$ be $\zeta_{\Pi_k,s}$. The maximal demand coverage which can be achieved with scenario $s \in S$ is $\zeta_s^* = \max_{\Pi_k \in \Pi} \zeta_{\Pi_k, s}$, which is obtained by solving the DMCLP with scenario s. The problem which minimizes the worst case regret can be formulated as:

$$\min_{\Pi_k \in \Pi} \max_{s \in S} (\zeta_s^* - \sum_t \sum_i d_i^t x_{is}^t) \tag{14}$$

subject to :

$$x_{is}^t \leq \sum_j a_{ij} y_{js}^t \qquad \forall i, t, s \tag{15}$$

$$y_{js}^t \geq y_{js}^{(t-1)} \qquad \forall j, t > 1, s \tag{16}$$

$$\sum_j y_{js}^t - y_{js}^{(t-1)} = p_s^t \qquad \forall t, s \tag{17}$$

$$\sum_{j'} R_{j'j} = 1 \qquad \forall j \tag{18}$$

$$\sum_j R_{j'j} = 1 \qquad \forall j' \tag{19}$$

$$\sum_{j'} j' R_{j'j} \leq \sum_{t' \leq t} p_s^{t'} + n(1 - y_{js}^t) \qquad \forall j, t, s \tag{20}$$

$$\sum_{j'} j' R_{j'j} \geq \sum_{t' \leq t} p_s^{t'} + 1 - n y_{js}^t \qquad \forall j, t, s \tag{21}$$

$$x_{is}^t, y_{js}^t, R_{j'j} = \{0, 1\} \tag{22}$$

In the above formulation constraint (15),(16) and (17) come from the DMCLP formulation. Constraint set (18) and (19) necessitate that all the facilities get a unique rank in a facility opening permutation. Constraint set (20) and (21) related the variables $R_{j'j}$ and y_{js}^t as explained earlier. Here the objective is nonlinear which can be linearized using the standard technique. The linear model for multi-period uncapacitated location problem with server uncertainty (MULPSU) can be formulated a follows:

$$\text{MULPSU:} \quad \text{Minimize } \theta \tag{23}$$

subject to :

$$x_{is}^t \leq \sum_j a_{ij} y_{js}^t \qquad \forall i, t, s \tag{24}$$

$$y_{js}^t \geq y_{js}^{(t-1)} \qquad \forall j, t > 1, s \tag{25}$$

$$\sum_j y_{js}^t - y_{js}^{(t-1)} = p_s^t \qquad \forall t, s \tag{26}$$

$$\sum_{j'} R_{j'j} = 1 \qquad \forall j \tag{27}$$

$$\sum_j R_{j'j} = 1 \qquad \forall j' \tag{28}$$

$$\sum_{j'} j' R_{j'j} \leq \sum_{t' \leq t} p_s^{t'} + n(1 - y_{js}^t) \qquad \forall j, t, s \tag{29}$$

$$\sum_{j'} j' R_{j'j} \geq \sum_{t' \leq t} p_s^{t'} + 1 - n y_{js}^t \qquad \forall j, t, s \tag{30}$$

$$\theta \geq (\zeta_s^* - \sum_t \sum_i d_i^t x_{is}^t) \qquad \forall s \tag{31}$$

$$x_{is}^t, y_{js}^t, R_{j'j} = \{0, 1\} \tag{32}$$

In the above formulation constraint set (31) has been used for making the model linear.

4 Solution methodology

4.1 Dominance rules for sequence of opening facilities

Let $\Pi_1 = (\pi_1, \pi_2, ...\pi_i, ...\pi_j, ...\pi_n)$ be a sequence in which facilities are to be opened, and let $\Pi_2 = (\pi_1, \pi_2, ...\pi_j, ...\pi_i, ...\pi_n)$ be a sequence obtained by switching the positions of π_i and π_j in Π_1. Let the demand at period t within the covering distance of a facility π_i be $d_{\pi_i}^t$, and total demand covered by the set of facilities $(\pi_1, \pi_2, ..., \pi_i)$ be $d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_i}^t$.

Proposition 1: If all the inequalities below are satisfied, facility opening sequence Π_1 will dominate Π_2, or in other words, regret associated with Π_1 will be not be higher than regret associated with Π_2 for any corresponding scenario.

$$\begin{aligned} d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_i}^t &\geq d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_j}^t && \forall t \in T \\ d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_i \cup \pi_{i+1}}^t &\geq d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_j \cup \pi_{i+1}}^t && \forall t \in T \\ &. \\ &. \\ d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_i \cup \pi_{i+1} \cup ... \cup \pi_{j-1}}^t &\geq d_{\pi_1 \cup \pi_2 \cup ... \cup \pi_j \cup \pi_{i+1} \cup ... \cup \pi_{j-1}}^t && \forall t \in T \end{aligned}$$

Proof. Assume that Π_1 does not dominate Π_2 even though all the above relationships are satisfied. Then there must exist a scenario of server availability $s = (a_1, a_2, ..a_T) \in S$ for which regret with Π_1 is more than regret with Π_2, or in other words demand coverage with Π_1 is less than the demand coverage with Π_2. If this hold then there must be atleast one period for which demand covered in that period with Π_1 is less than the demand coverage with Π_2.

$$d_{\pi_1 \cup \pi_2 \cup ... first\ (a_1 + a_2 + ..a_t)\ facilities\ in\ \Pi_1}^t < d_{\pi_1 \cup \pi_2 \cup ... first\ (a_1 + a_2 + ..a_t)\ facilities\ in\ \Pi_2}^t$$

for some $t \in T$. This contradicts the assumption that all the relations in the proposition are satisfied. Hence, Π_1 will dominate Π_2 if all the above set of relationships are satisfied. □

This dominance calculation will take $O(n^2 m|T|)$ time for each permutation and if a permutation is found to be dominated the time saved will be $O(nm|T||S|)$ which will be $O(n^5 m|T|)$ when $|T|$ is 5 periods.

4.2 Dominance rules for server availability scenarios

4.2.1 All the scenarios which have no new servers available in first or the last period

Proposition 2a: For any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, the regret in the server availability scenario $s_1 = (0, a_2, a_3, .., a_T) \in S$ will not be greater than regret in scenario $s_2 = (1, a_2 - 1, a_3, ..a_T) \in S$ if:

$$d_{\pi_1^k}^1 \leq LB(\zeta_{s_2}^*) - UB(\zeta_{s_1}^*) \tag{33}$$

where, $a_2 > 0$ and $d^1_{\pi^k_1}$ is the coverage by the first facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period of the planning horizon. Hence, it is the additional demand covered when the first facility of the facility opening sequence Π_k is opened in the first period instead of the second period. LB and UB represent the lower and upper bounds respectively.

Proof.

$$\begin{aligned} & d^1_{\pi^k_1} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \\ & \Rightarrow d^1_{\pi^k_1} \leq \zeta^*_{s_2} - \zeta^*_{s_1} && (34) \\ & \Rightarrow \zeta_{\Pi_k, s_2} - \zeta_{\Pi_k, s_1} \leq \zeta^*_{s_2} - \zeta^*_{s_1} && (35) \\ & \Rightarrow \zeta^*_{s_1} - \zeta_{\Pi_k, s_1} \leq \zeta^*_{s_2} - \zeta_{\Pi_k, s_2} && (36) \end{aligned}$$

or, Regret in scenario $s_1 = (0, a_2, a_3, .., a_T) \leq$ Regret in the scenario $s_2 = (1, a_2 - 1, a_3, ..a_T)$ □

Proposition 2b: If $a_2 = 0$ in scenario $s_1 = (0, a_2, a_3, .., a_T)$ and a_i be the first period with a non-zero element, for any facility opening sequence $\Pi_k = (\pi^k_1, \pi^k_2, ...\pi^k_n)$, regret in the scenario $s_1 = (0, .., a_i, .., a_T)$ will not be greater than regret in scenario $s_2 = (1, .., a_i - 1, .., a_T)$ if:

$$d^1_{\pi^k_1} + d^2_{\pi^k_1} + .. + d^{i-1}_{\pi^k_1} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \quad (37)$$

where, LHS is the coverage by the first facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period through $i - 1$ periods of the planning horizon. Hence, it is the additional demand covered when the first facility of the facility opening sequence Π_k is opened in the first period instead of the i^{th} period.

Proof. The proof for this is similar to the earlier one. Note that in this case

$$\zeta_{\Pi_k, s_2} - \zeta_{\Pi_k, s_1} = d^1_{\pi^k_1} + d^2_{\pi^k_1} + .. + d^{i-1}_{\pi^k_1} \quad (38)$$

□

Proposition 3a: For any facility opening sequence $\Pi_k = (\pi^k_1, \pi^k_2, ...\pi^k_n)$, regret in the scenario $s_1 = (a_1, a_2, .., a_{T-1}, 0)$ will not be greater than regret in scenario $s_2 = (a_1, a_2, .., a_{T-1} - 1, 1)$ if:

$$d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1} \cup \pi^k_n} - d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1}} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \quad (39)$$

where, $a_{T-1} > 0$ and LHS is the unique coverage by the last facility of any facility opening sequence $\Pi_k \in \Pi$ in the second last period of the planning horizon.

Proof.

$$\begin{aligned} & d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1} \cup \pi^k_n} - d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1}} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \\ & \Rightarrow d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1} \cup \pi^k_n} - d^{T-1}_{\pi^k_1 \cup \pi^k_2 \cup ... \cup \pi^k_{n-1}} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (40) \\ & \Rightarrow \zeta_{\Pi_k, s_1} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (41) \\ & \Rightarrow \zeta^*_{s_2} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta_{\Pi_k, s_1} && (42) \end{aligned}$$

or, Regret in the scenario $s_1 = (a_1, a_2, .., a_{T-1} - 1, 1) \geq$ Regret in scenario $s_2 = (a_1, a_2, .., a_{T-1}, 0)$ □

Proposition 3b: If $a_{T-1} = 0$ in scenario $s_1 = (a_1, .., a_{T-1}, 0)$ and a_i be the last period with a non-zero element, for any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (a_1, .., a_i, ., 0)$ will not be greater than regret in scenario $s_2(a_1, .., a_i - 1, ., 1)$ if:

$$\begin{aligned}(d^{T-1}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k\cup\pi_n^k} - d^{T-1}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k}) + (d^{T-2}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k\cup\pi_n^k} - d^{T-2}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k}) + ...\\ +(d^{i}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k\cup\pi_n^k} - d^{i}_{\pi_1^k\cup\pi_2^k\cup...\cup\pi_{n-1}^k}) \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2})\end{aligned} \tag{43}$$

where, LHS is the unique demand covered by the last facility of any facility opening sequence $\Pi_k \in \Pi$ in the i^{th} period through $(T-1)^{th}$ periods of the planning horizon. Hence, it is the additional demand covered when the last facility of the facility opening sequence Π_k is opened in the i^{th} period instead of the last period.

Proof. The proof for this is similar to the earlier one. Note that in this case the LHS is given by $\zeta_{\Pi_k,s_2} - \zeta_{\Pi_k,s_1}$. □

All possible facility opening permutation need not be checked to establish the dominance of the scenarios. One of the n facilities will be the last facility or first facility and thus checking for these n facilities in the LHS of the proposition 3 or 4 will establish the dominance. This dominance calculation will take $O(n^2m|T|)$ time for each scenario and if a scenario is found to be dominated the time saved will be $O(n^3m|T||Iter.|)$, where $|Iter.|)$ is the total number of iterations in the neighborhood search. Number of scenarios which have no new servers available in first or the last period is $2 \times \binom{n+|T|-2}{n} - \binom{n+|T|-3}{n}$. For example, with $n = 10$ and $|T| = 5$, number of such scenarios is 506 (out of 1001 total possible scenarios).

4.2.2 All the scenarios which have one new servers available in first or the last period

Proposition 4a: For any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (1, a_2, a_3, .., a_T)$ will not be greater than regret in scenario $s_2 = (2, a_2 - 1, a_3, ..a_T)$ if:

$$d^1_{\pi_1^k\cup\pi_2^k} - d^1_{\pi_1^k} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \tag{44}$$

where, $a_2 > 0$ and LHS is the unique coverage by the second facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period of the planning horizon. Hence, it is the additional demand covered when the second facility of the facility opening sequence Π_k is opened in the first period instead of the second period.

Proof.

$$d^1_{\pi_1^k\cup\pi_2^k} - d^1_{\pi_1^k} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1})$$

$$\Rightarrow d^1_{\pi_1^k\cup\pi_2^k} - d^1_{\pi_1^k} \leq \zeta^*_{s_2} - \zeta^*_{s_1} \tag{45}$$

$$\Rightarrow \zeta_{\Pi_k,s_2} - \zeta_{\Pi_k,s_1} \leq \zeta^*_{s_2} - \zeta^*_{s_1} \tag{46}$$

$$\Rightarrow \zeta^*_{s_1} - \zeta_{\Pi_k,s_1} \leq \zeta^*_{s_2} - \zeta_{\Pi_k,s_2} \tag{47}$$

or, Regret in scenario $s_1 = (1, a_2, a_3, ..a_T) \leq$ Regret in the scenario $s_2 = (2, a_2 - 1, a_3, ..a_T)$ □

Proposition 4b: If $a_2 = 0$ in scenario $(1, a_2, a_3, .., a_T)$ and a_i be the first period with a non-zero element(other than period 1), for any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (1, .., a_i, .., a_T)$ will not be greater than regret in scenario $s_2 = (2, .., a_i - 1, .., a_T)$ if:

$$(d^1_{\pi_1^k\cup\pi_2^k} - d^1_{\pi_1^k}) + (d^2_{\pi_1^k\cup\pi_2^k} - d^2_{\pi_1^k}) + .. + (d^{i-1}_{\pi_1^k\cup\pi_2^k} - d^{i-1}_{\pi_1^k}) \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \tag{48}$$

where, LHS is the unique coverage by the second facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period through $i-1$ periods of the planning horizon. Hence, it is the additional demand covered when the second facility of the facility opening sequence Π_k is opened in the first period instead of the i^{th} period.

Proof. The proof for this is similar to the earlier one. □

Proposition 5a: For any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (a_1, a_2, .., a_{T-1}, 1)$ will not be greater than regret in scenario $s_2 = (a_1, a_2, .., a_{T-1} - 1, 2)$ if:

$$d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \tag{49}$$

where, $a_{T-1} > 0$ and LHS is the unique coverage by the second last facility of any facility opening sequence $\Pi_k \in \Pi$ in the second last period of the planning horizon.

Proof.

$$\begin{aligned}
& d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \\
& \Rightarrow d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (50) \\
& \Rightarrow \zeta_{\Pi_k, s_1} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (51) \\
& \Rightarrow \zeta^*_{s_2} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta_{\Pi_k, s_1} && (52)
\end{aligned}$$

or, Regret in the scenario $s_2 = (a_1, a_2, .., a_{T-1} - 1, 2) \geq$ Regret in scenario $s_1 = (a_1, a_2, .., a_{T-1}, 1)$ □

Proposition 5b: If $a_{T-1} = 0$ in scenario $(a_1, .., a_{T-1}, 1)$ and a_i be the last period with a non-zero element(other than period T), for any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (a_1, .., a_i, ., 1)$ will not be greater than regret in scenario $s_2 = (a_1, .., a_i - 1, ., 2)$ if:

$$\begin{aligned}
(d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k}) + (d^{T-2}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{T-2}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k}) + ... \\
+ (d^{i}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k \cup \pi_{n-1}^k} - d^{i}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-2}^k}) \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2})
\end{aligned} \tag{53}$$

where, LHS is the unique demand covered by the second last facility of any facility opening sequence $\Pi_k \in \Pi$ in the i^{th} period through $(T-1)^{th}$ periods of the planning horizon.

Proof. The proof for this is similar to the earlier one. □

It can be seen from the proposition 4 and 5 that all possible facility opening permutation need not be checked to establish the dominance of the scenario. One of the n facilities will be the first (last) facility and the second (second last) facility will be one of the remaining $n-1$ facilities. Thus checking for these $n(n-1)$ possibilities in the LHS of the proposition 4 or 5 will establish the dominance. This dominance calculation will take $O(n^3 m|T|)$ time for each scenario and if a scenario is found to be dominated the time saved will be $O(n^3 m|T||Iter.|)$. Number of scenarios which have one new server available in first or the last period (excluding the scenarios which have no new server available in first or last period) is $2 \times \binom{n+|T|-3}{n-1} - \binom{n+|T|-5}{n-2} - 2 \times \binom{n+|T|-4}{n-1}$. For example, with $n = 10$ and $|T| = 5$, number of such scenarios is 285 (out of 1001 total possible scenarios).

4.2.3 All the scenarios which have two new servers available in first or the last period

Proposition 6a: For any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (2, a_2, a_3, .., a_T)$ will not be greater than regret in scenario $s_2 = (3, a_2 - 1, a_3, ..a_T)$ if:

$$d^1_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^1_{\pi_1^k \cup \pi_2^k} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \tag{54}$$

where, $a_2 > 0$ and LHS is the unique coverage by the third facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period of the planning horizon.

Proof.

$$\begin{aligned}
& d^1_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^1_{\pi_1^k \cup \pi_2^k} \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \\
& \Rightarrow d^1_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^1_{\pi_1^k \cup \pi_2^k} \leq \zeta^*_{s_2} - \zeta^*_{s_1} && (55) \\
& \Rightarrow \zeta_{\Pi_k, s_2} - \zeta_{\Pi_k, s_1} \leq \zeta^*_{s_2} - \zeta^*_{s_1} && (56) \\
& \Rightarrow \zeta^*_{s_1} - \zeta_{\Pi_k, s_1} \leq \zeta^*_{s_2} - \zeta_{\Pi_k, s_2} && (57)
\end{aligned}$$

or, Regret in scenario $s_1 = (2, a_2, a_3, ..a_T) \leq$ Regret in the scenario $s_2 = (3, a_2 - 1, a_3, ..a_T)$ □

Proposition 6b: If $a_2 = 0$ in scenario $(2, a_2, a_3, .., a_T)$ and a_i be the first period with a non-zero element(other than period 1), for any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (2, .., a_i, .., a_T)$ will not be greater than regret in scenario $s_2 = (3, .., a_i - 1, .., a_T)$ if:

$$(d^1_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^1_{\pi_1^k \cup \pi_2^k}) + (d^2_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^2_{\pi_1^k \cup \pi_2^k}) + .. + (d^{i-1}_{\pi_1^k \cup \pi_2^k \cup \pi_3^k} - d^{i-1}_{\pi_1^k \cup \pi_2^k}) \leq LB(\zeta^*_{s_2}) - UB(\zeta^*_{s_1}) \tag{58}$$

where, LHS is the unique coverage by the third facility of any facility opening sequence $\Pi_k \in \Pi$ in the first period through $i - 1$ periods of the planning horizon.

Proof. The proof for this is similar to the earlier one. □

Proposition 7a: For any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (a_1, a_2, .., a_{T-1}, 2)$ will not be greater than regret in scenario $s_2 = (a_1, a_2, .., a_{T-1} - 1, 3)$ if:

$$d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \tag{59}$$

where, $a_{T-1} > 0$ and LHS is the unique coverage by the third last facility of any facility opening sequence $\Pi_k \in \Pi$ in the second last period of the planning horizon.

Proof.

$$\begin{aligned}
& d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k} \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2}) \\
& \Rightarrow d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (60) \\
& \Rightarrow \zeta_{\Pi_k, s_1} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta^*_{s_2} && (61) \\
& \Rightarrow \zeta^*_{s_2} - \zeta_{\Pi_k, s_2} \geq \zeta^*_{s_1} - \zeta_{\Pi_k, s_1} && (62)
\end{aligned}$$

or, Regret in the scenario $s_2 = (a_1, a_2, .., a_{T-1} - 1, 3) \geq$ Regret in scenario $s_1 = (a_1, a_2, .., a_{T-1}, 2)$ □

Proposition 7b: If $a_{T-1} = 0$ in scenario $(a_1, .., a_{T-1}, 2)$ and a_i be the last period with a non-zero element(other than period T), for any facility opening sequence $\Pi_k = (\pi_1^k, \pi_2^k, ...\pi_n^k)$, regret in the scenario $s_1 = (a_1, .., a_i, ., 1)$ will not be greater than regret in scenario $s_2 = (a_1, .., a_i - 1, ., 2)$ if:

$$\begin{aligned}(d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{T-1}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k}) + (d^{T-2}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{T-2}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k}) + ... \\ +(d^{i}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k \cup \pi_{n-2}^k} - d^{i}_{\pi_1^k \cup \pi_2^k \cup ... \cup \pi_{n-3}^k}) \geq UB(\zeta^*_{s_1}) - LB(\zeta^*_{s_2})\end{aligned} \tag{63}$$

where, LHS is the unique demand covered by the third last facility of any facility opening sequence $\Pi_k \in \Pi$ in the i^{th} period through $(T-1)^{th}$ periods of the planning horizon.

Proof. The proof for this is similar to the earlier one. □

One of the n facilities will be the first (last) facility, the second (second last) and the third (third last) facility will be one of the remaining $n-1$ and $n-2$ facilities respectively. Thus checking for these $n(n-1)(n-2)$ possibilities in the LHS of the proposition 7 will establish the dominance. This dominance calculation will take $O(n^4 m|T|)$ time for each scenario and if a scenario is found to be dominated the time saved will be $O(n^3 m|T||Iter.|)$. Number of scenarios which have two new servers available in first or the last period (excluding the scenarios which have no/one new server available in first or last period) is $2 \times \binom{n+|T|-3}{n-1} - \binom{n+|T|-5}{n-2} - 2 \times \binom{n+|T|-4}{n-1} - 2 \times \binom{n+|T|-6}{n-3}$. For example, with $n = 10$ and $|T| = 5$, number of such scenarios is 140 (out of 1001 total possible scenarios).

In a similar fashion we can have dominance rules for three or higher number of servers available in the first (last) period. However, we have not used those relations because of two reasons. First, the computational time to check for dominance condition will be higher and as a result there will not be much saving in the time even if a scenario is found to be dominated. Secondly, number of scenarios, which have not been covered in the earlier propositions will be fewer. For example, with $n = 10$ and $|T| = 5$, a total of 931 out of 1001 all possible scenarios are covered in the earlier propositions.

4.3 Neighborhood search based methods

These methods start with an initial solution as the current solution and then check the neighborhood for a better solution. We have used the transposition neighborhood structure. In this neighborhood a move from one solution to the other can be executed by transposition of two distinct elements in the permutation.

4.3.1 Local search

Local search (LS) is one of the neighborhood search heuristics. It starts with a given initial solution as the current solution and checks its neighborhood for a better solution. If such a solution exist, the best neighbor will be selected as the current solution for the next iteration. If the neighborhood of the current solution does not contain any solution better than it, local search returns the current solution and terminates. This method may give a local optima as a solution and does not guarantee globally optimal solutions to most combinatorial problems. However, for many problems it returns relatively good quality solutions. The pseudocode for our local search implementation is given in algorithm 1.

4.3.2 Tabu search

Tabu search (TS) is one of the most effective improvements in the local search. TS starts with a given initial solution as the current solution. It performs a search on the neighbors of the current

Algorithm 1 Local search

```
Input: Co-ordinates, initial demand, growth rate of demand nodes; list of facilities and the covering
    distance
Output: Permutation of facility opening with minimax regret of demand coverage, and maximum
    regret associated with this permutation
Code
find all the inter-demand node distances
determine which nodes are within the covering distance of a candidate facility location
generate all facility opening scenarios
find the maximal demand coverage for each scenario by solving the multi-period maximal cov-
    erage location problem using CPLEX
generate initial permutation $\Pi$ of facility opening sequence (decreasing order of the total demand
    covered over the planning horizon)
find coverage and regret for $\Pi$ across each scenarios
set $\Pi^* \leftarrow \Pi$; $minimax\_reg^* \leftarrow$ maximum regret with $\Pi$ across scenarios
set $improv \leftarrow 1$
while $improv > 0$ do
    set $improv \leftarrow 0$; $tmp\_regret\_neigh \leftarrow \infty$
    generate all neighbors $\Pi_N$ of $\Pi$
    for each neighbor $\Pi_n$ of $\Pi$ do
        if $\Pi$ dominates $\Pi_n$ then
            go to next neighbor
        else
            find out the maximum regret across scenarios for $\Pi_n$
            if maximum regret across scenarios for $\Pi_n$ is less than $tmp\_regret\_neigh$ then
                set $\Pi^{nbr} \leftarrow \Pi_n$; $tmp\_regret\_neigh \leftarrow$ maximum regret across scenarios for $\Pi_n$
            end if
        end if
    end for
    if $tmp\_regret\_neigh$ is less than $minimax\_reg^*$ then
        $\Pi^* \leftarrow \Pi^{nbr}$; $minimax\_reg^* \leftarrow tmp\_regret\_neigh$
        set $improv \leftarrow 1$; $\Pi \leftarrow \Pi^{nbr}$ for the next iteration
    end if
end while
output $\Pi^*$ and $minimax\_reg^*$
```

solution and selects the next solution keeping the search history in mind. This technique allows it to overcome the local optima which local search cannot escape. A tabu list is maintained which prohibits certain moves from the current solution. A tabu tenure specifies the number of iterations a given move will be in the tabu list. It has been shown in the literature that random tabu tenure gives a better performance (Taillard, 1991, 1995). Another important feature of the tabu search is the aspiration criterion, which ensures that moves which are exceptionally promising are not ignored due to their tabu status. In our implementation the tabu tenure was a random number between 3 and 8. We used a simple aspiration criterion: a move is said to satisfy the aspiration criterion if it results in a solution with a regret lower than that of the best solution found thus far. A move satisfying the aspiration criteria can be selected, overruling its tabu status.

We give three implementations of the TS. In the first implementation (TS 1) we do not use any of the dominance rules. In the second implementation (TS 2), dominance rule for facility opening sequences has been considered. Third implementation (TS 3) includes all the dominance rules given in the earlier part of this section. In TS 3 a set of scenarios $S_0 \subset S$, which are not dominated by another scenario given in proposition 2 through 7, is found out. All the dominated scenarios

$S \backslash S_0$ are removed from further calculations in the algorithm. The pseudocodes for the three TS implementations are given in algorithm 2, 3 and 4.

Algorithm 2 Tabu search 1

Input: Co-ordinates, initial demand, growth rate of demand nodes; list of facilities, covering distance, tabu tenure parameter t, total no. of iterations k

Output: Permutation of facility opening with minimax regret of demand coverage, and maximum regret associated with this permutation

Code

```
find all the inter-demand node distances
determine which nodes are within the covering distance of a candidate facility location
generate all facility opening scenarios
find the maximal demand coverage for each scenario by solving the multi-period maximal coverage location problem using CPLEX
generate initial permutation $\Pi$ of facility opening sequence (decreasing order of the total demand covered over the planning horizon)
find coverage and regret for $\Pi$ across each scenarios
set $\Pi^* \leftarrow \Pi$; $minimax\_reg^* \leftarrow$ maximum regret with $\Pi$ across scenarios
for iterations from 1 to k do
    generate all neighbors $\Pi_N$ of $\Pi$
    for each neighbor $\Pi_n$ of $\Pi$ do
        if $\Pi_n$ is a non-tabu neighbor $\Pi_n$ then
            set $tmp\_regret\_neigh \leftarrow \infty$
            find out the maximum regret across scenarios for $\Pi_n$
            if maximum regret across scenarios for $\Pi_n$ is less than $tmp\_regret\_neigh$ then
                set $\Pi^{non-tabu\_nbr} \leftarrow \Pi_n$
                set $tmp\_regret\_neigh \leftarrow$ maximum regret across scenarios for $\Pi_n$
            end if
        else
            set $tabu\_regret\_neigh \leftarrow \infty$
            find out the maximum regret across scenarios for $\Pi_n$
            if maximum regret across scenarios for $\Pi_n$ is less than $tabu\_regret\_neigh$ then
                set $\Pi^{tabu\_nbr} \leftarrow \Pi_n$
                set $tabu\_regret\_neigh \leftarrow$ maximum regret across scenarios for $\Pi_n$
            end if
        end if
    end for
    select the best neighbor $\Pi^{nbr}$ keeping a check on the aspiration criterion
    update the tabu list using tabu tenure t selected randomly between an upper and a lower bound
    if maximum regret of $\Pi^{nbr}$ across all scenarios is less than $minimax\_reg^*$ then
        $\Pi^* \leftarrow \Pi^{nbr}$; $minimax\_reg^* \leftarrow$ maximum regret with $\Pi^{nbr}$ across scenarios
    end if
    set $\Pi \leftarrow \Pi^{nbr}$ for the next iteration
end for
output $\Pi^*$ and $minimax\_reg^*$
```

Algorithm 3 Tabu search 2 (with dominance criteria for facility opening sequences)

Input: Co-ordinates, initial demand, growth rate of demand nodes; list of facilities, covering distance, tabu tenure parameter t, total no. of iterations k

Output: Permutation of facility opening with minimax regret of demand coverage, and maximum regret associated with this permutation

Code

```
find all the inter-demand node distances
determine which nodes are within the covering distance of a candidate facility location
generate all facility opening scenarios
find the maximal demand coverage for each scenario by solving the multi-period maximal coverage location problem using CPLEX
generate initial permutation $\Pi$ of facility opening sequence (decreasing order of the total demand covered over the planning horizon)
find coverage and regret for $\Pi$ across each scenarios
set $\Pi^* \leftarrow \Pi$; $minimax\_reg^* \leftarrow$ maximum regret with $\Pi$ across scenarios
for iterations from 1 to k do
    generate all neighbors $\Pi_N$ of $\Pi$
    for each neighbor $\Pi_n$ of $\Pi$ do
        if $\Pi$ dominates $\Pi_n$ then
            go to next neighbor
        else
            if $\Pi_n$ is a non-tabu neighbor $\Pi_n$ then
                set $tmp\_regret\_neigh \leftarrow \infty$
                find out the maximum regret across scenarios for $\Pi_n$
                if maximum regret across scenarios for $\Pi_n$ is less than $tmp\_regret\_neigh$ then
                    set $\Pi^{non-tabu\_nbr} \leftarrow \Pi_n$
                    set $tmp\_regret\_neigh \leftarrow$ maximum regret across scenarios for $\Pi_n$
                end if
            else
                set $tabu\_regret\_neigh \leftarrow \infty$
                find out the maximum regret across scenarios for $\Pi_n$
                if maximum regret across scenarios for $\Pi_n$ is less than $tabu\_regret\_neigh$ then
                    set $\Pi^{tabu\_nbr} \leftarrow \Pi_n$
                    set $tabu\_regret\_neigh \leftarrow$ maximum regret across scenarios for $\Pi_n$
                end if
            end if
        end if
    end for
    if there are no undominated non-tabu neighbors of $\Pi$ then
        find the best dominated non-tabu neighbor using the above procedure
    end if
    select the best neighbor $\Pi^{nbr}$ keeping a check on the aspiration criterion
    update the tabu list using tabu tenure t selected randomly between an upper and a lower bound
    if maximum regret of $\Pi^{nbr}$ across all scenarios is less than $minimax\_reg^*$ then
        $\Pi^* \leftarrow \Pi^{nbr}$; $minimax\_reg^* \leftarrow$ maximum regret with $\Pi^{nbr}$ across scenarios
    end if
    set $\Pi \leftarrow \Pi^{nbr}$ for the next iteration
end for
output $\Pi^*$ and $minimax\_reg^*$
```

Algorithm 4 Tabu search 3 (with dominance criteria for facility opening sequences and scenarios)

Input: Co-ordinates, initial demand, growth rate of demand nodes; list of facilities, covering distance, tabu tenure parameter t, total no. of iterations k

Output: Permutation of facility opening with minimax regret of demand coverage, and maximum regret associated with this permutation

Code

```
find all the inter-demand node distances
determine which nodes are within the covering distance of a candidate facility location
generate all facility opening scenarios
find the maximal demand coverage for each scenario by solving the multi-period maximal coverage location problem using CPLEX
using scenario dominance rule find a set of scenarios $S_0 \subset S$ which are not dominated
generate initial permutation $\Pi$ of facility opening sequence (decreasing order of the total demand covered over the planning horizon)
find coverage and regret for $\Pi$ for each scenario $s_0 \in S_0$
set $\Pi^* \leftarrow \Pi$; $\mathit{minimax\_reg}^* \leftarrow$ maximum regret with $\Pi$ across $S_0$
for iterations from 1 to k do
    generate all neighbors $\Pi_N$ of $\Pi$
    for each neighbor $\Pi_n$ of $\Pi$ do
        if $\Pi$ dominates $\Pi_n$ then
            go to next neighbor
        else
            if $\Pi_n$ is a non-tabu neighbor $\Pi_n$ then
                set $\mathit{tmp\_regret\_neigh} \leftarrow \infty$
                find out the maximum regret across scenarios for $\Pi_n$
                if maximum regret across $S_0$ for $\Pi_n$ is less than $\mathit{tmp\_regret\_neigh}$ then
                    set $\Pi^{non-tabu\_nbr} \leftarrow \Pi_n$
                    set $\mathit{tmp\_regret\_neigh} \leftarrow$ maximum regret across $S_0$ for $\Pi_n$
                end if
            else
                set $\mathit{tabu\_regret\_neigh} \leftarrow \infty$
                find out the maximum regret across $S_0$ for $\Pi_n$
                if maximum regret across $S_0$ for $\Pi_n$ is less than $\mathit{tabu\_regret\_neigh}$ then
                    set $\Pi^{tabu\_nbr} \leftarrow \Pi_n$
                    set $\mathit{tabu\_regret\_neigh} \leftarrow$ maximum regret across $S_0$ for $\Pi_n$
                end if
            end if
        end if
    end for
    if there are no undominated non-tabu neighbors of $\Pi$ then
        find the best dominated non-tabu neighbor using the above procedure
    end if
    select the best neighbor $\Pi^{nbr}$ keeping a check on the aspiration criterion
    update the tabu list using tabu tenure t selected randomly between an upper and a lower bound
    if maximum regret of $\Pi^{nbr}$ across $S_0$ is less than $\mathit{minimax\_reg}^*$ then
        $\Pi^* \leftarrow \Pi^{nbr}$; $\mathit{minimax\_reg}^* \leftarrow$ maximum regret with $\Pi^{nbr}$ across $S_0$
    end if
    set $\Pi \leftarrow \Pi^{nbr}$ for the next iteration
end for
output $\Pi^*$ and $\mathit{minimax\_reg}^*$
```

5 Computational experiments

The procedures were coded in C++ (Visual Studio 2010). IBM ILOG CPLEX 12.4 was used as the IP solver. The procedures were implemented on an personal computer with Intel Core i5 (3.30 GHz) processors; 4 GB RAM; and windows 64-bit operating system.

Problems of size 5, 10 and 15 facilities with 100, 200 and 300 demand nodes were solved for $|T| = 5$ periods using CPLEX , local search (LS); and three tabu search implementations (TS1, TS2 and TS3). We generated abscissa and ordinates of all the demand nodes as random numbers from the uniform distribution [0, 100]. The first period demands of all the demand nodes was selected from a uniform distribution [200, 3000]. Annual demand growth rate which has been assumed to be deterministic and constant for each demand point, over the planning horizon has been generated from the uniform distribution [-0.04, 0.06]. Covering distance of 30 units was used with the problem instance of size 5 facilities, while for the problem size 10 and 15 facilities, a covering distance of 20 units was used.

We allowed our tabu search implementations to run for 1000 iterations. Further, a tabu tenure parameter was selected randomly from a uniform distribution [3, 8] in the TS implementations.

We generated fifty instances for the six problem sizes of 5 or 10 facilities with 100, 200 or 300 demand nodes. For the problem sizes of 15 facilities with 100, 200 or 300 demand nodes, only five instances were solved due to the large time taken for each run. The optimal solution for each of the instances in the problems was also found using the exact method (CPLEX 12.4) using the MULPSU formulation. We computed the gap of the solutions returned by LS and the three TS implementations with respect to the optimal solution. Gap for a heuristic solution θ_H compared to an optimal solution θ^* is defined as:

$$\text{Gap} = \theta_H - \theta^*$$

and has been given in the result tables whenever an exact solution could be calculated by CPLEX. We present the details of the performance of the solution methods on these problems in table 1 through 9.

It can be seen from the tables that for many instances local search did give the optimum solution, however an optimal solution is not guaranteed. TS 1 also could not give the optimal or the best solution for a few problems (table 4, 6 and 7), while the tabu search implementation TS 2 and TS 3 gave the optimal solution for all the instances which CPLEX could solve in reasonable time. This is because with the dominance rule on facility opening permutation, a set of non-optimal solution is cut off form the consideration set. This prevents the search getting stuck in local optima. TS 2 and TS 3 took significantly lower CPU time than the exact method. Time taken by TS 3 was the least in all the three tabu search implementations and at the same time it was able to find the best solution for all the instances.

One important observation is that the CPU time taken to find the optimal coverage for all the scenarios (by solving DMCLP using CPLEX) is the major component of the total CPU time taken by the algorithm. The neighborhood search after that takes relatively lesser time.

6 Conclusions

In this paper we have provided a formulation as well as solution method for the multi-period facility location problem with an uncertain number of servers, and the facility opening permutation as the decision variable. We study the performance of local search and three implementations of TS with a transposition neighborhood structure. The best customized tabu search algorithm was able to

Table 1: Computational results with 5 facilities and 100 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	10.140	34258	1.248	1863.2	13.852	0.0	3.135	0.0	1.794	0.0
2	8.143	0	1.123	0.0	14.867	0.0	3.931	0.0	1.528	0.0
3	7.488	0	1.154	0.0	14.567	0.0	4.602	0.0	1.625	0.0
4	9.048	16455.6	1.186	0.0	8.190	0.0	3.635	0.0	1.817	0.0
5	10.265	0	1.092	0.0	7.597	0.0	4.243	0.0	1.356	0.0
6	8.502	0	1.217	0.0	8.237	0.0	4.134	0.0	1.764	0.0
7	9.141	20565.9	1.123	0.0	8.455	0.0	3.697	0.0	1.821	0.0
8	6.662	0	1.076	0.0	8.674	0.0	3.978	0.0	1.617	0.0
9	8.611	1258.95	1.170	0.0	9.313	0.0	3.510	0.0	1.905	0.0
10	6.771	1529.74	1.092	0.0	9.438	0.0	3.401	0.0	1.622	0.0
11	6.099	0	1.092	0.0	10.483	0.0	4.649	0.0	1.643	0.0
12	9.407	4627.08	1.201	0.0	11.325	0.0	3.291	0.0	1.853	0.0
13	4.712	0	1.170	0.0	8.206	0.0	4.088	0.0	1.62	0.0
14	5.288	2167.61	1.154	0.0	12.605	0.0	3.853	0.0	1.608	0.0
15	4.773	0	1.170	0.0	14.037	0.0	4.118	0.0	1.598	0.0
16	4.368	178.917	1.155	0.0	14.534	0.0	4.212	0.0	1.491	0.0
17	3.370	0	1.123	0.0	14.028	0.0	3.978	0.0	1.636	0.0
18	3.697	0	1.139	0.0	14.061	0.0	4.368	0.0	1.636	0.0
19	4.196	5770.35	1.170	0.0	13.663	0.0	3.978	0.0	1.664	0.0
20	4.540	34258	1.154	1982.8	13.487	0.0	3.105	0.0	1.578	0.0
21	3.432	0	1.170	0.0	13.354	0.0	4.056	0.0	1.574	0.0
22	5.257	0	1.217	17543.8	14.352	0.0	4.024	0.0	1.68	0.0
23	6.911	3180.73	1.201	0.0	13.806	0.0	3.214	0.0	1.656	0.0
24	9.484	13738.7	1.217	0.0	14.555	0.0	3.307	0.0	1.468	0.0
25	5.492	0	1.185	0.0	15.304	0.0	4.399	0.0	1.374	0.0
26	7.550	781.67	1.217	0.0	14.009	0.0	3.245	0.0	1.372	0.0
27	4.960	0	1.123	0.0	12.667	0.0	3.276	0.0	1.611	0.0
28	4.400	0	1.139	0.0	12.652	0.0	3.885	0.0	1.576	0.0
29	5.600	0	1.170	0.0	12.979	0.0	3.822	0.0	1.666	0.0
30	5.007	5158.22	1.138	0.0	13.088	0.0	2.325	0.0	1.419	0.0
31	7.878	0	1.264	0.0	13.868	0.0	4.025	0.0	1.297	0.0
32	8.424	12318.4	1.248	0.0	13.042	0.0	3.416	0.0	1.602	0.0
33	9.501	2816.68	1.139	6322.5	13.400	0.0	3.603	0.0	1.911	0.0
34	8.221	3096.36	1.248	0.0	13.494	0.0	2.777	0.0	2.079	0.0
35	6.739	0	1.107	0.0	15.304	0.0	3.837	0.0	2.079	0.0
36	12.573	0	1.077	0.0	13.572	0.0	3.885	0.0	1.52	0.0
37	10.015	0	1.107	0.0	16.037	0.0	3.931	0.0	1.382	0.0
38	11.497	0	1.108	0.0	14.535	0.0	3.819	0.0	1.861	0.0
39	14.633	0	1.232	0.0	12.995	0.0	4.571	0.0	1.636	0.0
40	14.493	4314.38	1.279	25835.2	14.040	0.0	4.056	0.0	1.781	0.0
41	12.854	5393.51	1.076	15694.2	13.681	0.0	2.870	0.0	1.672	0.0
42	12.027	0	1.061	0.0	13.494	0.0	4.041	0.0	1.6	0.0
43	12.886	0	1.060	0.0	13.619	0.0	3.900	0.0	1.572	0.0
44	12.231	0	1.046	0.0	13.759	0.0	3.837	0.0	1.553	0.0
45	13.947	5047.65	1.123	0.0	14.539	0.0	3.713	0.0	1.995	0.0
46	10.935	0	1.029	0.0	13.260	0.0	3.354	0.0	1.458	0.0
47	12.730	0	1.046	0.0	12.090	0.0	3.573	0.0	1.543	0.0
48	12.885	0	1.107	0.0	11.185	0.0	4.118	0.0	1.543	0.0
49	12.761	0	1.030	0.0	13.759	0.0	3.775	0.0	1.505	0.0
50	13.510	0	1.154	0.0	15.210	0.0	4.150	0.0	1.598	0.0
Avg.	8.481		1.147		12.825		3.774		1.635	
Max.	14.633		1.279		16.037		4.649		2.079	

Table 2: Computational results with 5 facilities and 200 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	11.434	0	8.720	0.0	30.279	0.0	12.745	0.0	2.841	0.0
2	12.511	758.89	10.889	0.0	28.486	0.0	11.481	0.0	3.14	0.0
3	22.652	22376.8	10.514	0.0	29.063	0.0	10.889	0.0	3.337	0.0
4	22.917	19605.7	9.781	0.0	28.736	0.0	13.291	0.0	3.231	0.0
5	11.341	7915.36	8.237	0.0	26.629	0.0	13.151	0.0	3.183	0.0
6	12.558	0	8.346	0.0	25.599	0.0	13.775	0.0	2.929	0.0
7	13.557	0	7.785	0.0	26.380	0.0	11.373	0.0	2.738	0.0
8	11.700	0	8.471	0.0	26.364	0.0	11.061	0.0	2.756	0.0
9	15.990	0	9.220	0.0	26.396	0.0	13.806	0.0	3.006	0.0
10	16.068	37119.2	8.782	0.0	28.767	0.0	13.354	0.0	3.537	0.0
11	12.526	0	9.547	0.0	25.989	0.0	15.179	0.0	2.394	0.0
12	12.402	0	9.345	0.0	28.423	0.0	15.881	0.0	2.454	0.0
13	14.211	0	9.781	0.0	28.329	0.0	14.711	0.0	2.859	0.0
14	12.901	0	8.908	0.0	27.284	0.0	11.091	0.0	3.116	0.0
15	14.258	0	8.767	0.0	27.284	0.0	15.132	0.0	3.461	0.0
16	9.266	0	6.864	0.0	17.347	0.0	12.807	0.0	3.229	0.0
17	14.103	24160.2	7.254	0.0	18.970	0.0	13.806	0.0	3.588	0.0
18	9.188	17850.6	7.317	9659.1	19.235	0.0	15.351	0.0	3.128	0.0
19	10.967	0	7.504	0.0	19.875	0.0	20.686	0.0	3.34	0.0
20	12.121	0	7.394	0.0	24.320	0.0	18.861	0.0	3.315	0.0
21	12.511	124	7.020	0.0	19.905	0.0	17.238	0.0	3.106	0.0
22	29.468	179.876	6.630	0.0	26.717	0.0	14.383	0.0	2.955	0.0
23	8.767	0	6.646	0.0	29.006	0.0	21.341	0.0	3.136	0.0
24	13.182	0	6.380	0.0	30.325	0.0	20.576	0.0	3.14	0.0
25	20.483	23842.7	7.348	10155.5	25.509	0.0	19.843	0.0	3.398	0.0
26	14.321	35892.5	8.502	5841.9	26.537	0.0	13.650	0.0	3.68	0.0
27	13.167	0	7.534	0.0	29.453	0.0	14.399	0.0	3.593	0.0
28	21.450	3	4.774	0.0	29.616	0.0	14.633	0.0	3.672	0.0
29	15.101	29274.9	4.992	0.0	29.490	0.0	17.581	0.0	4.953	0.0
30	14.804	0	4.680	0.0	24.491	0.0	17.659	0.0	4.354	0.0
31	17.846	0	4.493	0.0	26.970	0.0	18.112	0.0	3.644	0.0
32	14.508	0	4.431	0.0	27.690	0.0	18.361	0.0	3.879	0.0
33	16.567	0	4.524	0.0	26.329	0.0	17.144	0.0	3.806	0.0
34	15.958	0	4.805	0.0	25.272	0.0	20.046	0.0	3.872	0.0
35	12.745	31860.2	4.555	0.0	25.771	0.0	13.868	0.0	3.74	0.0
36	11.154	0	4.431	0.0	27.05	0.0	17.394	0.0	3.957	0.0
37	9.891	0	4.040	0.0	33.447	0.0	19.843	0.0	3.989	0.0
38	11.170	0	4.836	0.0	34.898	0.0	18.626	0.0	4.184	0.0
39	15.382	0	4.649	0.0	30.888	0.0	20.389	0.0	3.816	0.0
40	16.973	0	4.025	0.0	31.122	0.0	14.898	0.0	3.846	0.0
41	14.009	0	4.024	0.0	24.914	0.0	18.174	0.0	4.276	0.0
42	30.249	41171.8	4.259	0.0	22.308	0.0	14.025	0.0	3.735	0.0
43	20.093	26193.5	4.227	0.0	21.466	0.0	21.106	0.0	3.542	0.0
44	8.549	0	4.103	0.0	17.534	0.0	18.377	0.0	3.216	0.0
45	8.050	0	3.978	0.0	19.172	0.0	17.597	0.0	3.272	0.0
46	5.585	0	4.088	0.0	19.594	0.0	18.003	0.0	3.797	0.0
47	32.791	7285.53	4.071	0.0	19.001	0.0	16.598	0.0	4.424	0.0
48	9.641	0	3.666	0.0	21.248	0.0	19.921	0.0	3.958	0.0
49	8.533	0	4.072	0.0	24.789	0.0	20.046	0.0	3.905	0.0
50	7.597	0	4.244	0.0	19.921	0.0	18.736	0.0	3.599	0.0
Avg.	14.464		6.469		25.684		16.220		3.481	
Max.	32.791		10.889		34.898		21.341		4.953	

Table 3: Computational results with 5 facilities and 300 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	19.174	0	10.249	0.0	18.205	0.0	14.515	0.0	4.744	0.0
2	16.967	0	11.294	0.0	17.534	0.0	12.559	0.0	4.616	0.0
3	27.541	0	10.936	0.0	17.254	0.0	11.014	0.0	4.919	0.0
4	24.482	0	10.686	0.0	18.486	0.0	11.894	0.0	4.945	0.0
5	26.039	0	10.202	0.0	19.095	0.0	11.630	0.0	3.812	0.0
6	12.574	69956.7	11.388	0.0	19.157	0.0	9.713	0.0	5.083	0.0
7	11.622	27816.5	10.171	0.0	19.172	0.0	6.925	0.0	4.295	0.0
8	14.773	448.258	10.983	0.0	18.439	0.0	7.385	0.0	4.542	0.0
9	32.058	33275.9	11.061	0.0	19.188	0.0	5.957	0.0	6.05	0.0
10	14.368	11258.4	10.718	0.0	19.624	0.0	7.258	0.0	6.005	0.0
11	14.617	0	10.920	0.0	20.920	0.0	10.319	0.0	5.591	0.0
12	13.384	0	10.374	0.0	20.514	0.0	10.358	0.0	6.152	0.0
13	16.567	0	10.655	0.0	20.436	0.0	10.818	0.0	5.806	0.0
14	14.492	0	10.514	0.0	19.890	0.0	10.925	0.0	5.783	0.0
15	19.640	0	10.093	0.0	20.171	0.0	10.642	0.0	6.149	0.0
16	25.771	0	10.234	0.0	19.734	0.0	11.023	0.0	6.08	0.0
17	13.354	0	10.546	0.0	19.672	0.0	11.219	0.0	5.475	0.0
18	22.495	53098.4	10.576	0.0	20.265	0.0	9.371	0.0	5.995	0.0
19	25.771	3221.21	10.405	0.0	20.187	0.0	8.823	0.0	5.891	0.0
20	21.871	0	10.998	0.0	20.093	0.0	11.650	0.0	5.388	0.0
21	32.073	0	9.782	0.0	20.592	0.0	9.654	0.0	5.504	0.0
22	32.246	45084.9	10.046	0.0	20.108	0.0	7.395	0.0	6.538	0.0
23	19.875	0	10.296	0.0	20.873	0.0	10.241	0.0	5.836	0.0
24	31.855	0	10.327	0.0	20.171	0.0	7.552	0.0	5.739	0.0
25	26.785	0	10.966	0.0	19.921	0.0	7.463	0.0	4.973	0.0
26	26.816	45587	10.998	0.0	20.311	0.0	7.629	0.0	6.482	0.0
27	17.114	0	10.577	0.0	20.030	0.0	8.715	0.0	4.549	0.0
28	28.673	4394.69	10.811	0.0	20.264	0.0	6.006	0.0	5.312	0.0
29	29.734	0	10.936	0.0	20.421	0.0	8.705	0.0	4.435	0.0
30	26.396	0	11.107	0.0	19.968	0.0	7.649	0.0	4.767	0.0
31	21.731	40575.8	11.201	4250.6	20.186	0.0	6.407	0.0	6.142	0.0
32	18.580	3088.02	10.452	0.0	19.625	0.0	5.780	0.0	7.487	0.0
33	9.282	0	10.389	0.0	20.014	0.0	7.600	0.0	4.851	0.0
34	14.836	0	11.482	0.0	20.514	0.0	8.011	0.0	5.195	0.0
35	13.619	0	11.232	0.0	19.953	0.0	8.520	0.0	5.048	0.0
36	14.462	0	10.343	0.0	19.859	0.0	8.177	0.0	5.28	0.0
37	11.997	0	10.499	0.0	19.844	0.0	8.372	0.0	4.311	0.0
38	14.727	0	11.154	0.0	19.983	0.0	7.864	0.0	6.186	0.0
39	17.597	0	10.468	0.0	19.828	0.0	7.864	0.0	6.171	0.0
40	17.971	0	10.935	0.0	20.296	0.0	8.451	0.0	5.954	0.0
41	13.088	0	10.842	0.0	19.718	0.0	8.314	0.0	5.781	0.0
42	46.847	4394.29	11.248	0.0	20.483	0.0	8.431	0.0	6.319	0.0
43	26.021	0	11.404	0.0	19.999	0.0	9.224	0.0	5.674	0.0
44	22.761	0	10.983	0.0	19.859	0.0	10.016	0.0	6.998	0.0
45	25.990	834.325	11.949	0.0	20.249	0.0	10.310	0.0	6.571	0.0
46	20.312	769.608	11.513	0.0	20.124	0.0	9.419	0.0	6.373	0.0
47	22.636	7948.93	11.669	0.0	20.795	0.0	9.938	0.0	6.547	0.0
48	39.578	9758.16	12.152	18667.2	20.405	0.0	9.625	0.0	6.639	0.0
49	17.020	0	11.903	0.0	19.640	0.0	9.527	0.0	5.314	0.0
50	30.701	0	12.028	0.0	20.217	0.0	11.062	0.0	6.401	0.0
Avg.	21.578		10.854		19.846		9.158		5.614	
Max.	46.847		12.152		20.920		14.515		7.487	

Table 4: Computational results with 10 facilities and 100 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	449.774	3989.44	73.616	0.0	304.730	0.0	133.800	0.0	76.021	0.0
2	132.990	0	70.980	0.0	304.933	0.0	314.403	0.0	85.876	0.0
3	236.699	0	70.715	0.0	324.745	0.0	212.581	0.0	102.928	0.0
4	241.894	21	69.639	0.0	325.867	0.0	134.020	0.0	87.603	0.0
5	291.567	0	68.344	0.0	286.371	0.0	227.246	0.0	100.879	0.0
6	168.634	343	70.184	0.0	327.382	0.0	114.051	0.0	63.999	0.0
7	162.577	0	69.529	13249.0	297.025	0.0	293.937	0.0	82.343	0.0
8	305.823	89	69.529	0.0	275.887	0.0	107.047	0.0	86.135	0.0
9	356.818	6862.13	67.782	0.0	326.618	0.0	115.393	0.0	79.812	0.0
10	506.549	5738.75	68.344	0.0	271.425	0.0	114.083	0.0	78.509	0.0
11	424.930	125.727	69.327	0.0	308.413	0.0	221.404	0.0	92.183	0.0
12	193.752	923.048	69.483	23033.8	273.188	0.0	124.787	0.0	84.41	0.0
13	159.573	0	68.359	0.0	280.738	0.0	195.172	0.0	99.551	0.0
14	287.914	0	69.811	0.0	271.893	0.0	227.199	0.0	102.491	0.0
15	607.122	9957.89	68.687	0.0	276.604	0.0	134.753	0.0	63.571	0.0
16	203.924	0	69.404	0.0	284.545	0.0	221.084	0.0	95.912	0.0
17	132.101	516.614	68.687	0.0	280.457	0.0	164.284	0.0	79.278	0.0
18	254.031	324.53	64.631	0.0	279.240	0.0	201.256	0.0	86.899	0.0
19	255.139	311	56.098	0.0	268.399	0.0	195.656	0.0	92.578	0.0
20	378.722	3989.44	57.471	0.0	263.126	0.0	124.332	0.0	69.497	0.0
21	299.841	232.184	55.911	0.0	262.798	0.0	208.042	0.0	81.469	0.0
22	179.476	1299.26	57.580	0.0	266.418	0.0	105.488	0.0	57.997	0.0
23	716.900	9261.36	57.362	0.0	265.481	0.0	89.856	0.0	80.91	0.0
24	157.591	40.8745	56.098	0.0	289.412	0.0	156.750	0.0	98.746	0.0
25	189.416	0	56.628	0.0	298.117	0.0	205.795	0.0	86.863	0.0
26	418.954	3572.88	56.379	4709.6	295.558	0.0	103.023	0.0	72.692	0.0
27	207.012	1989.63	56.426	0.0	341.172	0.0	189.025	0.0	119.066	0.0
28	204.750	18	54.631	0.0	286.682	0.0	178.480	0.0	72.555	0.0
29	132.444	0	56.254	0.0	307.960	0.0	255.934	0.0	84.393	0.0
30	171.818	5758.29	56.956	13586.0	289.958	0.0	113.193	0.0	91.499	0.0
31	375.227	451.181	57.128	0.0	337.117	0.0	182.458	0.0	105.096	0.0
32	260.692	11190.1	56.145	0.0	308.678	0.0	118.342	0.0	63.921	0.0
33	240.225	10876.1	56.051	3157.7	272.720	0.0	152.864	0.0	95.78	0.0
34	532.289	2662.87	56.128	0.0	266.308	0.0	84.583	0.0	76.144	0.0
35	237.823	3963.42	55.037	5664.8	278.023	0.0	157.638	0.0	85.272	0.0
36	288.834	46	56.378	0.0	270.910	0.0	219.250	0.0	77.71	0.0
37	139.823	8126.74	55.723	0.0	266.995	0.0	110.127	0.0	116.662	0.0
38	272.970	1851.47	56.066	0.0	269.397	0.0	122.912	0.0	73.953	0.0
39	319.426	0	55.302	0.0	270.692	0.0	212.878	0.0	98.038	0.0
40	190.882	2655	56.971	0.0	267.868	0.0	149.916	0.0	86.871	0.0
41	200.772	480.406	55.021	0.0	285.247	0.0	139.558	0.0	70.833	0.0
42	213.096	6.46165	55.022	0.0	282.704	0.0	164.050	0.0	78.685	0.0
43	216.997	416.384	56.441	14638.7	275.179	0.0	139.183	0.0	99.756	0.0
44	281.815	45.5691	55.473	0.0	278.220	0.0	102.586	0.0	64.023	0.0
45	707.040	5321.99	54.101	8133.8	258.945	14532.2	100.838	0.0	63.361	0.0
46	337.460	10113.2	54.444	0.0	250.864	0.0	135.689	0.0	83.035	0.0
47	166.374	941.92	55.864	0.0	262.876	0.0	231.208	0.0	83.911	0.0
48	173.737	0	58.000	0.0	263.983	0.0	285.044	0.0	95.602	0.0
49	299.240	0	57.018	0.0	261.005	0.0	171.116	0.0	98.709	0.0
50	279.209	224.832	57.018	0.0	261.940	0.0	98.451	0.0	93.259	0.0
Avg.	283.253		60.884		284.496		165.215		85.346	
Max.	716.900		73.616		341.172		314.403		119.066	

Table 5: Computational results with 10 facilities and 200 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	1012.145	220.432	92.367	0.0	538.403	0.0	180.804	0.0	139.961	0.0
2	597.746	0	114.255	0.0	549.542	0.0	331.453	0.0	197.216	0.0
3	716.275	415.349	117.999	0.0	559.402	0.0	165.454	0.0	141.284	0.0
4	795.524	8045.59	116.953	0.0	617.012	0.0	172.240	0.0	163.94	0.0
5	476.550	5310	118.451	0.0	616.451	0.0	199.867	0.0	170.963	0.0
6	835.741	4348.61	116.392	0.0	602.863	0.0	272.283	0.0	159.83	0.0
7	617.433	0	116.423	0.0	609.618	0.0	331.032	0.0	184.276	0.0
8	756.617	7045.41	116.611	0.0	639.368	0.0	249.694	0.0	195.707	0.0
9	852.307	0	114.598	0.0	587.013	0.0	332.530	0.0	211.404	0.0
10	1072.923	0	123.058	0.0	634.344	0.0	348.785	0.0	191.374	0.0
11	823.838	1691.32	143.107	0.0	584.939	0.0	184.392	0.0	191.374	0.0
12	332.655	0	143.677	0.0	600.897	0.0	381.207	0.0	164.204	0.0
13	953.936	3367.37	145.315	6218.7	591.990	0.0	163.405	0.0	164.204	0.0
14	2111.718	9501.81	143.271	0.0	622.488	0.0	141.873	0.0	128.357	0.0
15	1416.123	0	145.891	0.0	649.960	0.0	323.201	0.0	218.08	0.0
16	1288.626	2223.96	145.361	0.0	652.253	0.0	166.577	0.0	150.043	0.0
17	1487.338	11042	143.707	0.0	611.724	0.0	151.258	0.0	157.389	0.0
18	1205.461	72	139.886	0.0	567.436	0.0	169.931	0.0	167.47	0.0
19	1125.745	0	143.084	50481.9	923.738	0.0	319.644	0.0	208.021	0.0
20	1150.143	0	143.848	0.0	1138.334	0.0	299.146	0.0	201.472	0.0
21	1415.063	0	144.519	0.0	894.133	0.0	383.683	0.0	201.696	0.0
22	1068.119	0	142.069	42080.9	1214.837	0.0	547.966	0.0	178.713	0.0
23	1000.897	0	140.306	0.0	1186.336	0.0	331.001	0.0	179.258	0.0
24	3333.944	2541.86	146.360	0.0	836.910	0.0	290.551	0.0	194.541	0.0
25	2491.122	22428.9	140.041	0.0	966.843	0.0	206.825	0.0	118.043	0.0
26	970.244	2799.25	142.506	0.0	1095.527	0.0	216.778	0.0	137.21	0.0
27	1823.783	7453.7	139.683	12360.9	743.264	0.0	255.310	0.0	145.26	0.0
28	1062.767	1968.92	142.724	0.0	835.491	0.0	240.537	0.0	116.907	0.0
29	1597.724	3532.3	144.722	0.0	919.044	0.0	154.019	0.0	134.054	0.0
30	2369.316	2320.18	147.265	0.0	1108.928	0.0	357.787	0.0	147.436	0.0
31	1324.443	845.806	139.714	0.0	1115.697	0.0	279.537	0.0	157.89	0.0
32	2367.943	5125.91	139.636	252.0	1101.768	0.0	197.059	0.0	130.739	0.0
33	1006.140	1720.04	136.921	0.0	1448.837	0.0	210.273	0.0	118.152	0.0
34	1070.100	166	141.087	0.0	1238.720	0.0	344.418	0.0	155.847	0.0
35	825.788	6975.62	133.911	0.0	836.521	0.0	448.361	0.0	118.377	0.0
36	1164.354	16592.2	146.719	15760.2	1183.278	0.0	239.944	0.0	140.942	0.0
37	1970.050	11783.7	143.657	0.0	1145.853	0.0	234.999	0.0	112.466	0.0
38	1955.620	15296.8	143.657	0.0	976.733	0.0	246.823	0.0	129.873	0.0
39	1191.342	0	141.243	0.0	1247.877	0.0	438.642	0.0	165.753	0.0
40	1124.949	15826	139.964	0.0	1202.029	0.0	217.495	0.0	128.574	0.0
41	976.187	0	137.841	0.0	816.787	0.0	357.209	0.0	182.382	0.0
42	1413.003	18111	140.088	12021.3	1125.386	0.0	250.787	0.0	126.662	0.0
43	679.989	11353.9	140.977	0.0	1340.354	0.0	317.663	0.0	139.805	0.0
44	1209.829	1759.01	137.499	0.0	1106.744	0.0	246.590	0.0	159.614	0.0
45	1564.121	1697.56	144.596	0.0	1156.757	0.0	256.777	0.0	128.089	0.0
46	1454.656	170.245	142.475	0.0	1123.140	0.0	399.954	0.0	148.134	0.0
47	3717.596	17527.1	143.910	0.0	1122.531	0.0	268.242	0.0	134.397	0.0
48	1841.895	0	141.960	0.0	1056.761	0.0	416.817	0.0	177.5	0.0
49	1695.193	6631.2	140.041	0.0	1211.233	0.0	217.449	0.0	128.495	0.0
50	1401.818	4467.99	137.499	5596.3	941.477	0.0	167.841	0.0	128.275	0.0
Avg.	1334.337		136.557		903.951		272.522		157.433	
Max.	3717.596		147.265		1448.837		547.966		218.080	

Table 6: Computational results with 10 facilities and 300 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	1045.412	10248.9	180.320	0.0	1163.918	0.0	765.508	0.0	277.626	0.0
2	1323.182	70.9714	181.865	0.0	1000.585	0.0	712.188	0.0	250.297	0.0
3	4482.093	27414.5	181.241	0.0	1259.952	0.0	453.868	0.0	249.418	0.0
4	2088.002	11369.2	181.007	0.0	1161.859	0.0	633.860	0.0	230.281	0.0
5	2024.899	4927.08	182.474	0.0	1306.549	0.0	550.860	0.0	233.421	0.0
6	744.449	0	203.898	0.0	1258.142	0.0	866.644	0.0	312.202	0.0
7	1151.859	0	222.791	0.0	1189.923	0.0	851.175	0.0	302.719	0.0
8	1314.192	0	225.873	0.0	1600.532	0.0	806.955	0.0	316.177	0.0
9	3774.864	0	226.949	0.0	1228.783	0.0	774.418	0.0	302.595	0.0
10	2146.969	374.717	233.111	0.0	1226.645	0.0	572.090	0.0	256.703	0.0
11	2575.378	3639.42	220.288	0.0	1278.376	0.0	604.985	0.0	242.551	0.0
12	2646.826	351	229.819	0.0	1495.699	0.0	579.792	0.0	262.812	0.0
13	1218.721	0	225.234	0.0	1346.532	0.0	828.846	0.0	308.426	0.0
14	5255.322	22648.3	233.470	0.0	1321.454	0.0	440.954	0.0	238.905	0.0
15	1685.911	0	225.077	0.0	976.472	0.0	680.177	0.0	310.788	0.0
16	5651.749	20335.3	227.776	0.0	1034.469	0.0	472.415	0.0	208.853	0.0
17	1067.244	0	220.788	0.0	1232.669	0.0	867.081	0.0	262.744	0.0
18	659.913	12492.6	228.587	0.0	1343.350	0.0	600.664	0.0	225.716	0.0
19	933.474	724	225.031	0.0	1276.874	0.0	360.080	0.0	205.058	0.0
20	2483.789	0	226.200	0.0	1247.487	0.0	701.331	0.0	275.753	0.0
21	938.607	0	225.498	0.0	1258.049	0.0	634.485	0.0	254.429	0.0
22	1180.298	0	223.403	0.0	1247.582	0.0	887.220	0.0	267.568	0.0
23	2157.203	1857.83	224.381	0.0	1321.228	0.0	739.301	0.0	245.436	0.0
24	3381.290	2870.64	229.742	0.0	1475.512	0.0	384.088	0.0	207.726	0.0
25	1397.232	3819.54	214.017	0.0	1240.475	0.0	598.495	0.0	218.457	0.0
26	1221.076	0	232.721	0.0	1370.509	0.0	723.092	0.0	291.41	0.0
27	1149.270	34075.4	226.575	31499.7	1357.843	0.0	515.830	0.0	204.665	0.0
28	1715.769	747.918	223.580	0.0	1348.061	0.0	588.775	0.0	249.269	0.0
29	1715.410	21320	228.259	0.0	1019.415	0.0	699.614	0.0	210.369	0.0
30	1050.756	0	224.656	46192.2	1061.458	0.0	1183.121	0.0	290.705	0.0
31	1794.229	13962.2	214.064	0.0	1189.517	0.0	838.314	0.0	195.182	0.0
32	1394.130	0	172.989	10471.7	1073.767	0.0	762.389	0.0	250.367	0.0
33	1018.992	3252.46	173.800	0.0	994.939	0.0	675.606	0.0	184.116	0.0
34	1273.495	9.97008	174.518	0.0	1031.475	0.0	947.140	0.0	223.23	0.0
35	2504.987	0	192.114	0.0	1244.383	0.0	602.645	0.0	169.322	0.0
36	2409.627	13442.3	225.888	0.0	1299.374	0.0	661.675	0.0	185.734	0.0
37	2469.953	31803	228.229	21527.0	1397.765	21527.0	534.863	0.0	189.084	0.0
38	1098.959	0	220.647	0.0	1442.238	0.0	835.334	0.0	221.785	0.0
39	1211.482	0	229.694	0.0	1185.712	0.0	872.900	0.0	238.532	0.0
40	1184.525	0	210.975	0.0	954.285	0.0	1013.955	0.0	238.849	0.0
41	2274.843	2362.46	176.217	0.0	991.273	0.0	737.132	0.0	227.939	0.0
42	1199.393	0	174.283	0.0	1127.102	0.0	1257.705	0.0	225.888	0.0
43	879.014	342.725	188.339	0.0	1337.641	0.0	629.072	0.0	173.53	0.0
44	3747.954	19297.7	232.971	5773.0	1355.284	0.0	889.467	0.0	168.26	0.0
45	1071.441	0	223.891	0.0	1337.376	0.0	1224.914	0.0	235.714	0.0
46	1089.225	10929.9	218.323	0.0	1181.663	0.0	625.405	0.0	230.465	0.0
47	2004.354	7829.23	219.758	1904.0	1253.618	0.0	536.828	0.0	175.013	0.0
48	2496.832	397.435	215.593	0.0	1211.280	0.0	872.666	0.0	226.888	0.0
49	1053.064	0	174.128	0.0	1291.281	0.0	914.005	0.0	253.558	0.0
50	1515.123	13256.7	179.447	0.0	1329.575	0.0	315.883	0.0	180.594	0.0
Avg.	1877.456		211.610		1237.599		716.516		238.143	
Max.	5651.749		233.470		1600.532		1257.705		316.177	

Table 7: Computational results with 15 facilities and 100 demand nodes

	CPLEX		**LS**		**TS 1**		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap	CPU (s)	Gap
1	17816.530	3989.44	1,160.459	0.0	3882.051	0.0	1644.024	0.0	1315.222	0.0
2	4541.684	2740.07	1,189.393	5782.3	3891.333	5782.3	1540.753	0.0	1299.025	0.0
3	8855.220	2251.24	1,215.741	0.0	3984.964	0.0	1757.203	0.0	1548.522	0.0
4	15021.782	21	1,206.942	28883.6	3929.694	0.0	1603.402	0.0	1248.049	0.0
5	18196.776	4483.77	1,151.453	0.0	3767.157	0.0	1684.460	0.0	1548.755	0.0
Avg time	12886.398		1184.798		3891.040		1645.968		1391.915	
Max time	18196.776		1215.741		3984.964		1757.203		1548.755	

Table 8: Computational results with 15 facilities and 200 demand nodes

	LS		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Solution	CPU (s)	Solution
1	1,521.634	16133.67	2183.930	16133.67	1830.715	16133.67
2	2,084.943	13926.33	2084.215	13926.33	1653.9	13926.33
3	1,824.657	23076.71	2182.522	23076.71	1764.949	23076.71
4	2,000.079	29842.48	2332.360	9251.14	1696.491	9251.14
5	1,845.998	17133.97	2273.049	17133.97	1826.07	17133.97
Avg time	1855.462		2211.215		1754.425	
Max time	2084.943		2332.360		1830.715	

Table 9: Computational results with 15 facilities and 300 demand nodes

	LS		**TS 2**		**TS 3**	
Instance	CPU (s)	Solution	CPU (s)	Solution	CPU (s)	Solution
1	3,372.289	4942.97	3584.574	4942.97	3519.26	4942.97
2	3,382.102	41291.08	4338.742	33648	3620.82	33648
3	3,310.622	3044.44	5047.795	3044.44	3797.48	3044.44
4	3,224.994	0	11190.617	0	5872.22	0
5	3,022.958	26218.58	4268.118	23666	3598.61	23666
Avg time	3262.593		5685.969		4081.679	
Max time	3382.102		11190.617		5872.224	

solve instances of practical size which CPLEX could not solve. Furthermore, it gave the optimal solution for all the instances for which the optimal solution could be found using CPLEX and took significantly lesser time.

Future research direction can be to consider the problem when the facilities have capacity limitations. In that problem, all the operational facilities also need to considered, which was not the case in our current problem as all such facilities and the demand nodes catered by them could have been simply removed to obtain a reduced problem. Problems with a gradual covering model like discussed by Berman et al. (2003); Karasakal & Karasakal (2004); Berman et al. (2010) can also be considered where coverage function will be a non-increasing function of distance.

Acknowledgments

The author is thankful to Prof. Diptesh Ghosh and Prof. Sachin Jayaswal of IIM Ahmedabad for their guidance in this work.

References

Albareda-Sambola, M., Fernández, E., & Saldanha-da Gama, F. (2011). The facility location problem with bernoulli demands. *Omega*, *39*(3), 335–345.

Antunes, A., & Peeters, D. (2000). A dynamic optimization model for school network planning. *Socio-Economic Planning Sciences*, *34*(2), 101–120.

Antunes, A., & Peeters, D. (2001). On solving complex multi-period location models using simulated annealing. *European Journal of Operational Research*, *130*(1), 190–201.

Averbakh, I. (2003). Complexity of robust single facility location problems on networks with uncertain edge lengths. *Discrete Applied Mathematics*, *127*(3), 505–522.

Berman, O., Drezner, Z., & Krass, D. (2010). Generalized coverage: New developments in covering location models. *Computers & Operations Research*, *37*(10), 1675–1687.

Berman, O., Krass, D., & Drezner, Z. (2003). The gradual covering decay location problem on a network. *European Journal of Operational Research*, *151*(3), 474–480.

Berman, O., & Wang, J. (2011). The minmax regret gradual covering location problem on a network with incomplete information of demand weights. *European Journal of Operational Research*, *208*(3), 233–238.

Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*(3), 451–463.

Canel, C., Khumawala, B., Law, J., & Loh, A. (2001). An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Computers & Operations Research*, *28*(5), 411–427.

Chung, C.-H. (1986). Recent applications of the maximal covering location planning (MCLP) model. *Journal of the Operational Research Society*, 735–746.

Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers in Regional Science*, *32*(1), 101–118.

Church, R., & ReVelle, C. S. (1976). Theoretical and computational links between the p-median, location set-covering, and the maximal covering location problem. *Geographical Analysis*, *8*(4), 406–415.

Current, J., Ratick, S., & ReVelle, C. (1998). Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, *110*(3), 597–609.

Daskin, M. (2000). A new approach to solving the vertex p-center problem to optimality: Algorithm and computational results. *Communications of the Operations Research Society of Japan*, *45*(9), 428–436.

Dias, J., Captivo, M., & Clímaco, J. (2006). Capacitated dynamic location problems with opening, closure and reopening of facilities. *IMA Journal of Management Mathematics*, *17*(4), 317–348.

Dias, J., Captivo, M., & Clímaco, J. (2008). A memetic algorithm for multi-objective dynamic location problems. *Journal of Global Optimization*, *42*(2), 221–253.

Dias, J., Eugénia Captivo, M., & Clímaco, J. (2007). Efficient primal-dual heuristic for a dynamic location problem. *Computers & Operations Research*, *34*(6), 1800–1823.

Drezner, Z., & Hamacher, H. (2001). *Facility location: Applications and theory*. Springer.

Erlenkotter, D. (1981). A comparative study of approaches to dynamic location problems. *European Journal of Operational Research*, *6*(2), 133–143.

Farahani, R., & Hekmatfar, M. (2009). *Facility location: Concepts, models, algorithms and case studies*. Physica-Verlag HD.

Ghaderi, A., & Jabalameli, M. (2012). Modeling the budget-constrained dynamic uncapacitated facility location-network design problem and solving it via two efficient heuristics: A case study of health care. *Mathematical and Computer Modelling*.

Gunawardane, G. (1982). Dynamic versions of set covering type public facility location problems. *European Journal of Operational Research*, *10*(2), 190–195.

Hale, T. S., & Moberg, C. R. (2003). Location science research: A review. *Annals of Operations Research*, *123*(1-4), 21–35.

Karasakal, O., & Karasakal, E. K. (2004). A maximal covering location model in the presence of partial coverage. *Computers & Operations Research*, *31*(9), 1515–1526.

Killmer, K. A., Anandalingam, G., & Malcolm, S. A. (2001). Siting noxious facilities under uncertainty. *European Journal of Operational Research*, *133*(3), 596–607.

Lawn, J. E., Rohde, J., Rifkin, S., Were, M., Paul, V. K., & Chopra, M. (2008). Alma-ata 30 years on: Revolutionary, relevant, and time to revitalise. *The Lancet*, *372*(9642), 917–927.

Melo, M., Nickel, S., & Saldanha da Gama, F. (2006). Dynamic multi-commodity capacitated facility location: A mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research*, *33*(1), 181–208.

Rohde, J., Cousens, S., Chopra, M., Tangcharoensathien, V., Black, R., Bhutta, Z. A., & Lawn, J. E. (2008). 30 years after alma-ata: Has primary health care worked in countries? *The Lancet*, *372*(9642), 950–961.

Saldanha da Gama, F., & Captivo, M. (1998). A heuristic approach for the discrete dynamic location problem. *Location Science*, *6*(1), 211–223.

Schilling, D. A., Jayaraman, V., & Barkhi, R. (1993). A review of covering problems in facility location. *Computers & Operations Research*.

Taillard, E. (1991). Robust taboo search for the quadratic assignment problem. *Parallel computing*, *17*(4), 443–455.

Taillard, E. (1995). Comparison of iterative searches for the quadratic assignment problem. *Location science*, *3*(2), 87–105.

Van Roy, T., & Erlenkotter, D. (1982). A dual-based procedure for dynamic facility location. *Management Science*, *28*(10), 1091–1105.

Walley, J., Lawn, J. E., Tinker, A., De Francisco, A., Chopra, M., Rudan, I., … Black, R. E. (2008). Primary health care: Making alma-ata a reality. *The Lancet*, *372*(9642), 1001–1007.

Warszawski, A. (1973). Multi-dimensional location problems. *Operational Research Quarterly*, 165–179.

Wesolowsky, G., & Truscott, W. (1975). The multiperiod location-allocation problem with relocation of facilities. *Management Science*, *22*(1), 57–65.

Zarandi, M. H. F., Davari, S., & Sisakht, S. A. H. (2013). The large-scale dynamic maximal covering location problem. *Mathematical and Computer Modelling*, *57*(3), 710–719.