**Journal of Sciences, Islamic Republic of Iran 23(1): 67-75 (2012)** http://jsciences.ut.ac.ir
University of Tehran, ISSN 1016-1104

# A New Heuristic Solution Method for Maximal Covering Location-Allocation Problem with *M/M/1* Queueing System

F. Moeen Moghadas* and H. Taghizadeh Kakhki

*Department of Mathematics, Faculty of Mathematics and Statistics Science, Ferdowsi University of Mashhad, Vakil Abad Blvd., Mashhad 91775-1159, Islamic Republic of Iran*

Received: 20 November 2010 / Revised: 23 January 2011 / Accepted: 25 February 2012

## Abstract

We consider the queueing maximal covering location-allocation problem (QM-CLAP) with an *M/M/1* queueing system. We propose a new solution procedure based on decomposition of the problem into smaller sub-problems. We solve the resulting sub-problems both with a branch and bound algorithm and with the meta-heuristic GRASP. We also solve the entire model with GRASP. Computational results for these approaches are compared with the solutions obtained by CPLEX. Results show that using the new procedure in which sub-problems were solved with Branch and bound is better.

**Keywords:** Maximal covering location problems; *M/M/1* queueing system; Heuristics; GRASP

## Introduction

The maximal covering location problem (MCLP) has been modeled as a binary integer program by Church and ReVelle [8]. This problem seeks to maximize the coverage of demand points with a given number of facilities. Some extensions and variants of this problem, including deterministic and probabilistic models, as well as its application in emergency facility location, are discussed in many review papers including those of ReVelle [33]; Schilling et al. [36], Marianov and ReVelle [23], Brotcorne et al. [7], Marianov and Serra [26]; and Goldberg [18]. ReVelle et al [34] presented a bibliography for some discrete location problems, including maximal covering location problem.

The problem we are concerned with here, is what sometimes has been termed as location in congested systems. The setting for these problems is usually a queueing system representing a stochastic demand.

The first location model in congested system and with stochastic demand is the stochastic queue median (SQM) model with an *M/G/1* queueing system introduced by Berman et al [6]. Since then others have also tried to expand on this notion. Marianov and ReVelle [22] consider the queueing probabilistic location set covering problem with *M/G/s/s* queueing system. Later [23], they used an *M/G/s/s* queueing system for siting emergency facilities.

Marianov and Serra [24] consider probabilistic maximal covering models with constraints on waiting time, and for queue length. Wang et al [38] model the problem of locating automated teller machines (ATM's) as an *M/M/1* queueing system with the assumption that customers travel to the closest open facility, and with constraints placed on maximum waiting times at facilities. An extension for this problem with *M/M/k*

* Corresponding author, Tel.: +98(511)8828604, Fax: +98(511)8828604, E-mail: forooghmoeen@gmail.com

queueing system is considered by Berman and Drezner [4]. Marianov and Serra [25] present a set covering formulation of the problem and extend the models to cover all population. They assume an *M/M/m* queueing system at each facility, and reduce the model to a linear programming problem. Marianov [21] considers location of multiple servers in an *M/M/m* queueing system. Silva and Serra [37] considered MCLP with *M/M/1* queueing system and different priority levels. They formulate the problem and propose a heuristic procedure to solve this problem.

A review of the models with stochastic demands and congestion at facilities is given by Berman and Krass [5]. A computational comparison for five maximal covering models, namely, the MCLP, the maximal expected covering location problem (MEXCLP), MCLP plus probabilistic response time (PR), MEXCLP plus PR, and MEXCLP plus PR and station specific busy time is presented in Erkut et al. [12].

Moeen Moghadas and Taghizadeh Kakhki [29] consider the maximal covering location-allocation problem with an *M/M/k* queueing system. In this model, *k* is unknown and additional constraints on the number of servers at each center as well as constraints on the total costs of establishing a center and locating servers are imposed. They [30] also consider the queueing maximal covering problem with an *M/G/1* queueing system. In this problem a single mobile server resides at each center, and demands for service occur in time as a *Poisson* process. If the server is available, it is immediately dispatched to the demand point. After providing the service, the server returns to its base. If the server is busy, the customer waits in a queue with an *M/G/1* system.

In this paper we consider the queueing maximal covering location-allocation problem discussed in [24]. The objective is to choose the location of at most *p* service centers and allocate demand points to those centers so that the population covered is maximized. The servers are fixed and the customers must travel to centers to receive the service. If the server is busy, the customer enters a queue with an *M/M/1* system. In addition, since one of the indicators of the service quality is considered to be the average waiting time in queue at service centers, therefore in this model demand points are assigned to centers so that the average waiting time in each center does not exceed a predetermined amount. Application of this model can be found in, for example, finding appropriate locations for health centers, such as hospitals, and emergency medical units, as well as determining the locations of banks, police stations, and post offices.

In what follows we first discuss a mathematical formulation of the problem; then in section two we propose a new solution procedure which is based on relaxation and decomposition of the problem into smaller sub-problems. In sections three and four we use the meta-heuristic GRASP to solve these sub-problems and the entire problem. Finally, computational results are presented in section five.

### 1- Model Formulation

The model presented by Marianov and Serra [24], the queueing maximal covering location-allocation model (QM-CLAP), with the average waiting time at a center constrained to be less than a given time, is as follows:

$$P) Max \sum_{i \in I} \sum_{j \in N_i} a_i x_{ij}$$

$$s.t \quad x_{ij} \le y_j \qquad \forall i \in I, j \in J \qquad (1)$$

$$\sum_{j \in N_i} x_{ij} \le 1 \qquad \forall i \in I \qquad (2)$$

$$\sum_{j \in J} y_j \le p \qquad (3)$$

$$W_j \le \tau_j \qquad \forall j \in J \qquad (4)$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall i \in I, j \in J \qquad (5)$$

where,

I: The set of all existing demand points (incident locations) ($|I| = m$)

J: The set of all possible locations of new facilities (centers) ($|J| = n$)

$N_i$: The set of demand points in a pre-specified neighborhood of $i$; i.e., $N_i = \{j \in J : d(i,j) \le R\}$, where R is the covering radius, and $d(i,j)$ is the distance between node i and candidate center j

$y_j$ : is 1 if a new facility is located at site $j \in J$; and 0, otherwise

$x_{ij}$ : is 1 if a call from point i is answered by facility (center) $j$ ; and 0, otherwise

$a_i$ : Population at demand point i

p : The maximum number of new facilities (centers) $(p < n)$

$W_j$ : Average waiting time at facility (center) j

$\tau_j$ : Maximum allowable waiting time at center $j$ .

The objective maximizes the population covered. Constraints (1) ensure that a point is being served only by an established facility at $j$. Constraints (2) guarantee that each point i is to be allocated to at most one service center $j$; Constraint (3) establishes at most $p$ new facilities, and (4) ensure that the average waiting time at each center j does not exceed a predetermined amount $\tau_j$.

The above model is in fact a modification of the well known *p*-median problem, with constraint set (4) added, and with the objective being maximization of population covered.

To state the problem properly, however, we need an explicit form expressing the average waiting time, $W_j$, in terms of the variables $x_{ij}$ and $y_j$. This in turn, is system dependent.

If we assume that the arriving calls from a demand point i have a *Poisson* distribution with intensity $f_i$, then the arrival rate at each center $j$, $\lambda_j$, can be calculated by $\lambda_j = \sum_{i \in I : j \in N_i} f_i x_{ij}$ (see e.g. Marianov and Serra [24]). '$i \in I : j \in N_i$' presents the set of all demand points i which are in a given neighborhood of candidate facility $j$.

If the stability condition $(\lambda_j < \mu)$ holds; then the waiting time is equal to $W_j = \dfrac{\lambda_j}{\mu(\mu - \lambda_j)}$ and the constraints (4) can be replaced by;

$$\sum_{i \in I : j \in N_i} f_i x_{ij} \le \frac{\mu^2 \tau_j}{1 + \mu \tau_j} \qquad (4')$$

where $\mu$ is the service rate. We also assume an infinite capacity queue and a FIFO discipline. $\mu$ and $\tau_j$ are fixed and are assumed to be equal for all candidate centers.

Therefore the model with *M/M/1* queueing system is as follows:

$$P)\ Max \sum_{i \in I} \sum_{j \in N_i} a_i x_{ij}$$

$$s.t \quad (1)-(3), (4') \, and \, (5)$$

This problem is known to be NP-complete (Marianov and Serra [24]) and can only be solved using software packages such as CPLEX, for small problems. Other solution methods and heuristics such as greedy, lagrangian relaxation, column generation and heuristic concentration have also been used to solve simpler version of this problem; i.e., the maximal covering location problem (see e.g. Resende [32], Marianov and Serra [26], Pereira et al [31], ReVelle et al [35]).

A GRASP type heuristic for the priority queueing covering location problem (PQCLP) for an *M/M/1* system has been proposed by Silva and Serra [37]. Their first model with one priority in fact, corresponds to the problem considered here.

Galvao and Morabito [17] considered the use of hypercube queueing model in the solution of probabilistic location problems. Correa et al [10] proposed a constructive genetic algorithm for the model presented by Marianov and Serra [24]. In another study Correa et al [9] considered clustering search (CS) for this problem. They reported results of different methods applied to this problem and indicated CS got better results than others heuristics. Correa et al [11] present the QM-CLAP as a covering graph and then partition the graph using the graph partitioning heuristic METIS of Karypis and Kumar (1998) which results in a block diagonal structure representation of the QM-CLAP. A Dantzig-Wolfe decomposition procedure is then applied.

### *2- Heuristic Algorithm*

In this section we propose a new heuristic algorithm to solve the problem $P$. The idea for our proposed algorithm comes from the solution methods suggested in the literature for the capacitated *p*-median problem (see e.g., Baldacci et al [2], Lorena and Senne [20]).

To motivate the discussion, suppose that in an iterative fashion, we have determined the location of $k-1$ centers and want to determine the location of the $k^{th}$ center $(k \le p)$ from among the candidate locations. Then, if we remove the demand points covered by the previous $k-1$ centers, constraints (2) would be satisfied for any new facility. In addition if we ignore the constraints (1) and (3), and consider them only implicitly, then the problem can be decomposed into smaller knapsack sub-problems, which is then solved for each candidate facility $j$, as follows:

$$Knap_j)\ F_j = Max \sum_{i \in \bar{I} : j \in N_i} a_i x_{ij}$$

$$s.t \quad \sum_{i \in \bar{I} : j \in N_i} f_i x_{ij} \le \frac{\mu^2 \tau_j}{1 + \mu \tau_j}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in \bar{I}, j \in J$$

where, $\bar{I}$ is the set of demand points not covered by the $k-1$ previous centers.

Now suppose that we want to choose at most p centers from among n candidate locations. We start with $k=1$, solve $n-k+1$ knapsack sub-problems to find the optimal solution $j^*$; delete all the nodes assigned to $j^*$ from $\bar{I}$; Add $j^*$ to $J^*$, the set of selected facilities; Increment $k$, and continue until $k>p$, or all the nodes are covered. This procedure is outlined in Fig. (1).

*SolveKnap* $(j, \bar{I})$, solves the knapsack sub-problem $Knap_j$ for a candidate center $j$, and if a demand point $i\,(i \in \bar{I})$ is assigned to $j$; then it is added to $I_j^*$. *Update* $\left(j^*, J^*, \bar{I}\right)$ is the procedure for updating two sets $J^*$ and $\bar{I}$.

Notice that this procedure can be easily extended to solve the more general case of fixed charge facilities; i.e., when the objective is:

$$Max \sum_{i \in I\ :j \in N_i} a_i x_{ij} - \sum_{j \in J} c_j y_j$$

where $c_j$ is the cost of establishing a facility at site $j$.

Note also that a simple upper bound for the knapsack sub-problems can be easily obtained using Dantzig's theorem (see Martello and Toth [28] or Kellerer et al [19]) which can then be used as a starting solution for a branch and bound procedure, as has been done here.

Note also that when the calling rate is constant, we would have a special case of a knapsack problem; i.e., a subset sum problem (SSP), which is classified by Martello et al. [27] as the type of knapsack problems with bounded coefficients for which LP and ILP solutions are sufficiently close, and hence, can be solved "very fast".

In addition to the above procedure, we have tried to solve this problem using a GRASP type procedure. This procedure is also used by Silva and Serra [37] for their priority queueing model. There are, however, a few differences between our implementation and theirs, which will be explained in section 5.

### *3- GRASP Procedure for Solving Sub-Problem Knap$_j$*

The heuristic procedure GRASP (Greedy Randomized Adaptive Search Procedure) has been developed in late 1980's by Feo and Resende ([13], [14]). It consists of a construction phase and a local search phase. In the construction phase, a solution is built using a greedy function and randomization. The local search phase, then finds an optimal solution in the neighborhood of the solution obtained in the construction phase. A recent bibliography of GRASP for both algorithms and applications is presented by Festa and Resende ([15], [16]).

*Set* $k=1$, $J^*=\emptyset$, $\bar{I}=\{1,2,\dots,m\}$
*While* $k \le p$ & $\bar{I} \ne \emptyset$ *do*
  *For all* $j \notin J^*$ *do*
    $[F_j, I_j^*] = SolveKnap\,(j, \bar{I})$
  *End*
  $j^* = Arg\left(Max\,\{F_j : j \notin J^*\}\right)$
  $[J^*, \bar{I}] = Update\,(j^*, J^*, \bar{I})$
  $k = k+1$
*End*

**Figure 1.** Heuristic procedure.

*Function* $\left[I_j^*, F_j\right] = GRASP - SubProblem\,(\bar{I}, j)$
*Set* $k=1$, $key=1$, $I_j^*=\emptyset$
$BestSolution=\emptyset$, $W(BestSolution)=0$
*While* $k \le Maxitr$ & $key == 1$ *do*
  $\left[I_j^*, key\right] = ConstructGRS\_SubProblem\left(\bar{I}, I_j^*\right)$
  *If* $key == 1$ *then*
    $I_j^* = LocalSearch\_SubProblem(I_j^*)$
    *If* $W\left(I_j^*\right) > W(BestSolution)$ *then*
      $BestSolution = I_j^*$
    *End*
  *End*
  $k=k+1$
*End*
$I_j^* = BestSolution$
$F_j = W(BestSolution)$

**Figure 2.** GRASP algorithm for solving problem *Knap$_j$*.

*Function* $RCL\_SP = MakeRCL\_SubProblem(\bar{I}, I_j^*)$
$RCL\_SP=\emptyset$
$a^* = Max\,\{a_i : i \in \bar{I} \backslash I_j^*,\ S1+f_i \le \frac{\mu^2\tau_j}{1+\mu\tau_j}\}$
*For all* $i \in \bar{I}\backslash I_j^*$ *do*
  *If* $a_i \ge \alpha \times a^*$ & $S1+f_i \le \frac{\mu^2\tau_j}{1+\mu\tau_j}$ *then*
    $RCL\_SP = RCL\_SP \cup \{i\}$
  *End*
*End*

**Figure 3.** Procedure for constructing the restricted candidate list.

Here we present a GRASP procedure to solve sub-problem $Knap_j$. The outline of the algorithm is given in Fig. (2) as follow:

In this procedure $\bar{I}$, as before, is the set of demand points not yet assigned to a center, $I_j^*$ is the set of demand points assigned to center at $j$ and $W\left(I_j^*\right) = \sum_{i \in I_j^*} a_i$. The steps of the algorithm are repeated for a maximum of $Maxitr$ iterations, and the best solution is stored in $BestSolution$.

For our problem in the construction phase, a restricted candidate list (*RCL*) of demand points that can improve the objective while maintaining the waiting time condition, is constructed. If this list is not empty; then a candidate point is randomly selected from it, and $key$ is set equal to 1; else $key$ is set equal to 0 and no local search is performed. If $key = 1$ then the local search is performed and the $BestSolution$ is updated. Fig. (3) shows the procedure for constructing the restricted candidate list. In this procedure the so called candidate parameter $\alpha$ ranges from zero to one. $\alpha = 0$ indicates that the points are randomly selected, while $\alpha = 1$ yields the greedy selection. In addition we define $S1$ as $S1 = \sum_{i \in I_j^*} f_i$.

The neighborhood structure suggested by Resende [32] who used GRASP to solve the maximal covering location problem, is the 2-exchange neighborhood. Here, we also use the 2-exchange construct. The exchange between two demand points $i \in I_j^*$ and $s \in \bar{I} \setminus I_j^*$ is only possible if first of all $a_s > a_i$, and second, adding s to $I_j^*$ and deleting i from $I_j^*$ would not violate the constraints. The procedure for local search is shown in Fig. (4). In this procedure $S1$ is defined as before.

### *4- GRASP Algorithm for Solving Problem P*

In addition to solving the sub-problems with GRASP, we have also tried to solve problem P with this heuristic. The procedure is similar to that proposed for the maximal covering problem by Resende [32]. The main differences are in construction of the restricted candidate list, and the greedy function. The construction of the restricted candidate list is outlined in Fig. (5).

For any candidate facility j $(j \in J \setminus J^*)$ we use GRASP to solve the sub-problem $Knap_j$, as indicated in line 3.

We would like to point out here a few differences between our implementation of GRASP for this problem and that of Silva and Serra [37] for priority queues. In Silva and Serra [37] the allocation is based on the set $D_{ij}$, constructed for each candidate center j, which includes the indices of all demand points i, within reach (given distance) from center $j$, ordered according to their distances from $j$. Demand points are then, allocated to $j$ until the waiting time constraint is violated, similar to what has been done by Marianov and Serra [24]. We, on the other hand obtain this set by solving sub-problems. Another main difference is in the construction of the restricted candidate list (RCL). Their greedy function is the total costumer arrival rates, while ours is the objective function. Finally we use a 2-exchange neighborhood strategy, while they employ a comprehensive search over all feasible solutions which improve the objective.

```
Function I_j^* = LocalSearch_SubProblem ( I_j^* )
    key = 1
    While key == 1 do
        For any i ∈ I_j^* do
            For any s ∈ Ī\I_j^* do
                If ( S1 − f_i + f_s ≤ μ^2τ_j / (1+μτ_j) ) & a_s > a_i then
                    S1 = S1 − f_i + f_s
                    I_j^* = I_j^* ∪ {s}\{i}
                    Ī = Ī ∪ {i}\{s}
                    key = 0
                End
            End
        End
    End
```

**Figure 4.** Procedure for Local Search for the *Knap*$_j$ sub-problems.

```
Function RCL_MP = MakeRCL_MainProblem( J^* , Ī )
1. RCL_MP = ∅
2. For any j ∈ J\J^* do
3.     [ I_j^* , F_j ] = GRASP_SubProblem ( Ī , j )
4. End
5. F_max = Max { F_j : j ∈ J\J^* }
6. RCL_MP = { j : j ∈ J\J^* , F_j ≥ α × F_max }
```

**Figure 5.** Procedure for constructing the restricted candidate list for problem *P*.

**Table 1**. Computational results for test problems; Sub-problems were solved with a branch and bound algorithm

| Number of customers (demand points) | Percent of population covered | | | | | Least number of servers for 100% pop. coverage | Average CPU time for 100% pop coverage (seconds) |
|---|---|---|---|---|---|---|---|
| | *p*=3 | *p*=5 | *p*=10 | *p*=15 | *p*=20 | | |
| 20 | 100 | - | - | - | - | 3 | 0.01 |
| 30 | 98.53 | - | - | - | - | 4 | 0.02 |
| 50 | 74.57 | 85.80 | - | - | - | 9 | 0.03 |
| 100 | 66.48 | 79.65 | 96.06 | - | - | 14 | 0.10 |
| 150 | 61.60 | 73.40 | 92.44 | 99.13 | - | 17 | 0.21 |
| 200 | 56.74 | 71.54 | 92.80 | 98.31 | - | 18 | 0.32 |
| 324 | 22.11 | 35.16 | 60.07 | 76.78 | 87.18 | 37 | 1.28 |
| 818 | 8.96 | 14.93 | 29.86 | 44.79 | 59.73 | 43 | 28.45 |

**Table 2.** Computational results for test problems; Sub-problems were solved with GRASP

| Number of customers (demand points) | Percent of population covered | | | | | Least number of servers for 100% pop. coverage | Average CPU time for 100% pop. coverage (seconds) |
|---|---|---|---|---|---|---|---|
| | *p*=3 | *p*=5 | *p*=10 | *p*=15 | *p*=20 | | |
| 20 | 100 | - | - | - | - | 3 | 0.05 |
| 30 | 98.53 | - | - | - | - | 4 | 0.11 |
| 50 | 74.57 | 85.80 | - | - | - | 9 | 0.30 |
| 100 | 66.48 | 79.65 | 96.06 | - | - | 14 | 1.19 |
| 150 | 60.68 | 73.02 | 92.15 | 99.13 | - | 17 | 2.99 |
| 200 | 54.70 | 69.70 | 91.76 | 97.54 | - | 19 | 6.23 |
| 324 | 22.11 | 35.31 | 60.36 | 77.39 | 88.40 | 36 | 19.69 |
| 818 | 8.96 | 14.96 | 30.01 | 45.16 | 60.10 | 42 | 425.53 |

**Table 3**. Computational results for test problems; Problems were solved with GRASP

| Number of customers (demand points) | Percent of population covered | | | | | Least number of servers for 100% pop. coverage | Average CPU time for 100% pop. coverage (seconds) |
|---|---|---|---|---|---|---|---|
| | *p*=3 | *p*=5 | *p*=10 | *p*=15 | *p*=20 | | |
| 20 | - | - | - | - | - | 2 | 5.48 |
| 30 | 98.61 | - | - | - | - | 4 | 13.04 |
| 50 | 75.20 | 87.42 | - | - | - | 9 | 24.52 |
| 100 | 66.62 | 80.28 | 96.91 | - | - | 14 | 93.21 |
| 150 | 60.68 | 74.58 | 94.51 | 99.71 | - | 16 | 208.89 |
| 200 | 55.89 | 71.77 | 92.80 | 98.94 | - | 17 | 393.22 |
| 324 | 22.12 | 35.32 | 60.40 | 77.48 | 88.69 | 36 | 1483.25 |
| 818 | 8.96 | 14.93 | 29.84 | 44.73 | 59.58 | 42 | 24727.34 |

**Table 4**. Comparison of the results with CPLEX and the results with heuristic algorithms

| Number of customers | Least number of servers for 100% population coverage Using CPLEX software [CPU time] | Percent of population covered | | |
|---|---|---|---|---|
| | | sub- problems solved with branch and bound [CPU time] | Sub- problems solved with GRASP [CPU time] | GRASP for the entire problem [CPU time] |
| 20 | 2 [0.02] | 98.10 [0.01] | 98.10 [0.05] | 100 [5.48] |
| 30 | 4 [0.02 ] | 100 [0.02] | 100 [0.11] | 100 [13.04] |
| 50 | 9 [0 .04] | 100 [0.03] | 100 [0.30] | 100 [24.52] |
| 100 | 13 [0 .66] | 99.29 [0.09] | 99.29 [1.18] | 99.78 [92.75] |
| 150 | 14 [ 29.33] | 98.62 [0.21] | 98.24 [2.92] | 99.41 [206.18] |
| 200 | 15 [ 1.08] | 98.31 [0.28] | 97.54 [5.82] | 98.94 [379.26] |
| 324 | 29 [ 12.20] | 97.24 [1.03] | 97.85 [19.17] | 97.94 [1229.35] |
| 818 | 37* [8836.76] | 99.08 [26.57] | 99.37 [417.49] | 99.76 [24653.36] |

*- The smallest number we were able to obtain a solution for!

### *5- Computational Results*

We have tried to solve some test problems, both adapted from the literature, and randomly generated, by the proposed algorithms. All calculations were performed on a Pentium IV processor with 2.80 GHz and 2.50 GB of RAM.

To test the efficiency of the algorithm, we have solved some test problems. The number of points in the samples ranged from 20 to 200. The 30 point problem is that of Marianov and Serra [24]; others are generated randomly. The 324 and 818 node problems are from OR-Lib (Beasely [3]) for the capacitated *p*-median problem, which are also available at http://www.lac.inpe.br/~lorena/instancias.html for queueing maximal covering location-allocation problems, and are used by Correa et al ([9] and [11]), too.

In these examples the distances are considered to be Euclidean, and the number of candidate locations, $n$, was taken to be equal to the number of points, $m$. We assume that each demand point is also a potential server location. $f_i$, $\tau_j$ and $\mu$ are the same as the ones used in Silva and Serra [37]; namely, the daily call rate of 0.005 times the population, an average time limit of 12.75 minutes, the average service time of 10 minutes; The covering radii $R$ were taken to be 1.5 miles for 20-200 point problems, 250 km for 324 point problem, and 750 km for 818 point problem.

In Table 1 below the results for the test problems solved with the branch and bound [1] algorithm with Dantzig's upper bounds for the sub-problems, are presented. In Table 2 the results when the sub-problems are solved with GRASP are presented. The number of iterations and parameter $\alpha$ were set to 50 and 0.85, respectively.

Table 3 shows the results when the entire model P is solved with GRASP.

In order to be able to make a reasonable comparison, we have solved our test problems with CPLEX, too. The results obtained with ILOG-CPLEX 10.2 under Unix system, are shown in Table 4. Average CPU times (seconds) are shown in brackets ([ ]).

It is evident from these tables that as the problem size increases we have a wider gap between the optimal solution and the heuristic solution; even when the optimal is known, the heuristics do not produce the same results due to the limit imposed on the number of iterations. On the other hand, the solutions obtained by CPLEX do not exhibit a uniform behavior in terms of CPU time for different problem sizes, as well as for different number of new facilities for a given problem size. For example, for the 818 point problem, for $p=38$ it took 51579.4 seconds to obtain the optimal, for $p=37$ it took 8836.76 seconds, as indicated, while for $p = 36$ we were not able to get a solution in 141450 seconds! This is while we have a rather uniform behavior for the heuristics!

## Results and Discussion

In this paper we considered the maximal covering problem in a congested system with an *M/M/1* queueing system. A heuristic procedure based on decomposition of the problem into smaller knapsack sub-problems was proposed. In addition we solved the resulting sub-problems both with a branch and bound algorithm and with GRASP. We also solved the entire model with GRASP. Computational results for these approaches were compared with the solutions obtained by CPLEX. The results indicate that coverage percentages are the same for the three variants of the solution method. However, as the running times, especially for the large scale problems, are much less for the first method, using branch and bound for the sub-problems should be preferred.

## References


1. Balas E. An additive algorithm for solving linear programming in zero-one variables. Operations Research, **13**(4): 517-546 (1965).
2. Baldacci R., Hadjiconstantinou E., Maniezzo V., and Mingozzi A. A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research*, **29**: 365-386 (2002).
3. Beasley J.E. OR-Library: distributing test problems by electronic mail. Journal of the Operational Research Society, **41**: 1069-1072 (1990).
4. Berman O., and Drezner Z. The multiple server location problem. Journal of the Operational Research Society, **58**: 91-99 (2007).
5. Berman O., and Krass D. Facility location problems with stochastic demands and congestion. In: Drezner Z., and Hamacher H.W. (Eds), Facility Location: Application and Theory. Springer, pp. 329-371 (2004).
6. Berman O., Larson R.C., and Chiu S. S. Optimal server location on a network operating as an M/G/1 queue. Operations Research, **33** (4): 746-771 (1985).
7. Brotcorne L., Laporte G., and Semet F. Ambulance location and relocation models. European Journal of Operational Research, **147**: 451–463 (2003).
8. Church R., and ReVelle C. The maximal covering location problem. Papers of the Regional Science Association, **32**: 101-118 (1974).
9. Corrêa F.D.A., Chaves A.A., and Lorena L.A.N. Hybrid heuristics for the probabilistic maximal covering location-allocation problem. Operational Research, An International Journal **7**(3): 323-344 (2008).
10. Corrêa F.D.A., and Lorena L.A.N. Using the constructive genetic algorithm for solving the probabilistic maximal covering location-allocation problem. In: Workshop on computational intelligence/SBRN. Available at http://www.lac.inpe.br/~lorena/correa/Correa_Lorena_Wci_2006.pdf. (2006).
11. Corrêa F.D.A., Lorena L.A.N., and Ribeiro G.M. A decomposition approach for the probabilistic maximal covering location-allocation problem. *Computers & Operations Research*, **36** (10): 2729-2739 (2009).
12. Erkut E., Ingolfsson A., Sim T., and Erdoğan G. Computational comparison of five maximal covering models for locating ambulances. Geographical Analysis, **41**: 43-65 (2009).
13. Feo T.A., and Resende M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, **8**: 67–71 (1989).
14. Feo T.A., and Resende M.G.C. Greedy randomized adaptive search procedures. Journal of Global Optimization, **6**: 109-133 (1995).
15. Festa P., and Resende M.G.C. An annotated bibliography of GRASP – Part I: Algorithms. International Transactions in Operational Research, **16**: 1-24 (2009).
16. Festa P., and Resende M.G.C. An annotated bibliography of GRASP – Part II: Applications. International Transactions in Operational Research, **16**: 131-172 (2009).
17. Galvao R.D., and Morabito R. Emergency service systems: The use of the hypercube queueing model in the solution of probabilistic location problems. International Transactions in Operational Research, **15**: 525-549 (2008).
18. Goldberg J.B. Operations research models for the deployment of emergency services vehicles. EMS Management Journal, **1**(1): 20-39 (2004)
19. Kellerer H., Pferschy U., and Pisinger D. Knapsack problems. Springer (2004).
20. Lorena L.A.N., and Senne E.L.F. Local search heuristics for capacitated p-median problems. Networks and Spatial Economics, **3**(4): 407-419 (2003).
21. Marianov V. Location of multiple-server congestible facilities for maximizing expected demand, when services are non-essential. Annals of Operations Research, **123**: 125–141 (2003).
22. Marianov V., and ReVelle C. The queuing probabilistic location set covering problem and some extensions. Socio-Economic Planning Science, **28**: 167-178 (1994).
23. Marianov, V., and ReVelle, C. The queueing maximal availability location problem: A model for the siting of emergency vehicles. European Journal of Operational Research, **93**: 110-120 (1996).
24. Marianov V., and Serra D. Probabilistic, maximal covering location-allocation models for congested systems. Journal of Regional Science, **38**(3): 401-424 (1998).
25. Marianov V., and Serra D. Location–allocation of multiple-server service centers with constrained queues or waiting times. Annals of Operations Research, **111**: 35–50 (2002).
26. Marianov V., and Serra D. Location problems in the public sector. In: Drezner Z., and Hamacher H.W. (Eds), Facility location: Application and theory. Springer, pp.119-150 (2004).
27. Martello S., Pisinger D., and Toth P. New trends in exact algorithms for the 0-1 knapsack problem. European Journal of Operational Research, **123**: 325-332 (2000).

28. Martello S., and Toth P. Knapsack problems, algorithms and computer implementations. John Wiley & Sons (1990).
29. Moeen Moghadas F., and Taghizadeh Kakhki H., Maximal covering location-allocation problem with M/M/k queueing system and side constraints. Iranian Journal of Operations Research **2**(2): 1-16 (2011)
30. Moeen Moghadas F., and Taghizadeh Kakhki H., Queueing maximal covering location-allocation problem: an extension with M/G/1 queueing systems. Advances in Decision Sciences, doi: 10.1155/2011/605629 (2011).
31. Pereira M.A., Lorena L.A.N., and Senne E.L.F. A column generation approach for the maximal covering location problem. International Transactions in Operational Research, **14**: 349-364 (2007).
32. Resende M.G.C. Computing approximate solutions of the maximum covering problem with GRASP. Journal of Heuristics, **4**: 161-171 (1998).
33. ReVelle C. Review, extension and prediction in emergency service sitting models. European Journal of Operational Research, **40**: 58-69 (1989).
34. ReVelle C.S., Eiselt H.A., and Daskin M.S. A bibliography for some fundamental problem categories in discrete location science. European Journal of Operational Research, **184**: 817–848 (2008).
35. ReVelle C., Scholssberg M., and Williams J. Solving the maximal covering location problem with heuristic concentration. Computers & Operations Research, **35**: 427-435 (2008).
36. Schilling D.A., Jayaraman V., and Barkhi R. A review of covering problems in facility location. Location Science, **1**(1): 25-55 (1993).
37. Silva F., and Serra D. Locating emergency services with different priorities: the priority queuing covering location problem. Journal of the Operational Research Society, **59**(9): 1229-1238 (2008).
38. Wang Q., Batta R., and Rump C.M. Algorithms for a facility location problem with stochastic customer demand and immobile servers. Annals of Operations Research, **111**: 17-34 (2002).