Hindawi Publishing Corporation
Advances in Decision Sciences
Volume 2011, Article ID 605629, 13 pages
doi:10.1155/2011/605629

Research Article

Queueing Maximal Covering Location-Allocation Problem: An Extension with *M/G/1* Queueing Systems

Foroogh Moeen Moghadas and Hossein Taghizadeh Kakhki

Department of Mathematics, Ferdowsi University of Mashhad, Vakil Abad Boulevard,
Mashhad 91775-1159, Iran

Correspondence should be addressed to Foroogh Moeen Moghadas, fo_mo59@stu-mail.um.ac.ir

Received 8 January 2011; Revised 28 August 2011; Accepted 7 November 2011

Academic Editor: Roger Z. Ríos-Mercado

We consider the queueing maximal covering location-allocation problem (QM-CLAP) with an $M/G/1$ queueing system. We first formulate the problem as a binary quadratic programming problem and then propose a new solution procedure based on decomposition of the problem into smaller binary quadratic sub-problems. The heuristic procedure GRASP is used to solve the sub-problems, as well as the entire model. Some computational results are also presented.

1. Introduction

An extensively studied and widely used model in location theory is the maximal covering location problem (MCLP). It can be stated as follows. Given a set of m demand points and a set of n potential new facility locations in the plane, find the location of $p\ (< n)$ new facilities such that the demands covered are maximized. A demand point i is said to be covered by a facility j if the distance (or time) between i and j is no more than a given fixed value, R. In this simple form, the problem is attributed to Church and ReVelle [1] who have modeled it as a binary integer program. Extensions and applications of this model have been discussed by many including ReVelle [2], Schilling et al. [3], Marianov and ReVelle [4], Brotcorne et al. [5], Marianov and Serra [6], and Goldberg [7]. A bibliography for some discrete location problems, including maximal covering location problem, can also be found in ReVelle et al. [8]. Erkut et al. [9] present a computational comparison for five maximal covering models.

A variant of this problem, usually called location in congested systems, takes into account the stochastic nature of demand and response times. Berman et al. [10] are among the first to incorporate such systems into location models. They extend Hakimi's 1-median problem by assuming that demands at nodes follow a Poisson process, and the service time has a general distribution, hence, a network with an *M/G/1* queueing system. They propose an algorithm to find the optimal location of a server on such networks. Berman and Mandowsky [11] present a location-allocation model for districting a region with congestion. Berman et al. [12] consider the problem of locating a single facility with *M/G/k* queueing system. Batta and Berman [13] consider locating a single facility on a network with an *M/G/k* queue. Batta [14] considers the problem of locating a single server on a network operating as an *M/G/1* with nonpreemptive priority queue. Marianov and ReVelle [15] consider the queueing probabilistic location set covering problem with *M/G/s/s* queueing system. They [4], also use an *M/G/s/s* queueing system for siting emergency facilities. Marianov and Serra [16] consider probabilistic maximal covering models with constraints on waiting time and for queue length. Jamil et al. [17] consider the problem of locating a single facility with a "center-type" objective in an *M/G/1* queueing environment. Silva and Serra [18] consider the MCLP with an *M/M/1* queueing system and with different priority levels. They consider both a "directed choice environment" model, where a demand point may be allocated to one center for one priority and to another center for a different priority, and a "user choice environment," where it is assumed that the customer always chooses the closest center. They propose a heuristic procedure to solve this problem. Moghadas and Kakhki [19] consider the QM-CLAP with *M/M/k* system and side constraints. In this model, k is unknown and some 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. Berman and Krass [20] is a thorough review of the models with stochastic demands and congestion at facilities.

In this paper, we consider the maximal covering location problem in a congested system. 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. The objective is to choose the location of at most p service centers and to allocate demand points to those centers so that the population covered is maximized. Each demand point should be serviced by at most one established service center, and the average waiting time at each service center must not exceed a given threshold. We first formulate the problem as a binary quadratic programming problem with linear and quadratic constrains. Then we propose two solution procedures. The first procedure is based on a decomposition of the problem into smaller binary quadratic knapsack-type subproblems. The second method is the metaheuristic procedure GRASP. Finally some computational results are presented.

2. Problem Formulation

In order to formulate the problem, we use the following notations and variables:

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

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

–a_i: population at point i,

–N_i: the set of points in a pre-specified neighborhood of i; that is, $N_i = \{j \in J : d(i, j) \leq R\}$, R is the covering radius, and $d(i, j)$ is the distance between node i and candidate center j,

–p: the maximum number of new facilities (centers) ($p < n$),

–W_j: average waiting time at facility (center) j,

–τ_j: maximum allowable waiting time at center j,

$$x_{ij} = \begin{cases} 1, & \text{if a call from point } i \text{ is answered by facility (center) } j, \\ 0, & \text{otherwise,} \end{cases}$$
$$y_j = \begin{cases} 1, & \text{if a new facility is located at site } j \in J, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The queueing maximal covering location-allocation problem (QM-CLAP), with average waiting time at a center constrained to be less than a given time, is modeled as follows (Marianov and Serra [16]):

$$\text{Max} \quad \sum_{i \in I} \sum_{j \in N_i} a_i x_{ij}, \tag{P}$$

$$\text{s.t.} \quad \sum_{j \in N_i} x_{ij} \le 1, \quad \forall i \in I, \tag{2.2}$$

$$x_{ij} \le y_j, \quad \forall i \in I,\ j \in J, \tag{2.3}$$

$$\sum_{j \in J} y_j \le p, \tag{2.4}$$

$$W_j \le \tau_j, \quad \forall j \in J, \tag{2.5}$$

$$x_{ij}, y_j \in \{0,1\}, \quad \forall i \in I,\ j \in J. \tag{2.6}$$

Constraints (2.2) guarantee that each point i is allocated to at most one service center j. Constraints (2.3) ensure that a point is being served only by an established facility at j, Constraint (2.4) establishes at most p new centers, and (2.5) ensures that the average waiting time at each center j does not exceed a predetermined amount τ_j. The objective maximizes the population covered.

Marianov and Serra [16] considered this problem for the *M/M/1* and *M/M/m* queueing systems and proposed heuristic methods to solve them. Corrêa and Lorena [21] proposed a constructive genetic algorithm, and Corrêa et al. [22] proposed a clustering search procedure to solve this problem. Corrêa et al. [23] convert the problem into a covering graph problem and propose a decomposition approach to solve it. Silva and Serra [18] considered the priority queueing covering location problem (PQCLP) for an *M/M/1* system and proposed a GRASP heuristic procedure to solve their problem.

In this paper, we consider a problem in which requests for services at each demand point occur with a Poisson process. If the server is available at the time of a call, it travels to the demand point to provide on-scene and perhaps off-scene services. A service is completed only when the server returns home. If the server is unavailable, the customer is entered into a queue with an *M/G/1* system, with an infinite queue length and a FIFO discipline. If we assume that the arriving calls from a demand point i have a Poisson distribution with intensity f_i, then the requests for service at center j is a Poisson process with an intensity λ_j, $\lambda_j = \sum_{i\in I:j\in N_i} f_i x_{ij}$ (see e.g., Marianov and Serra [16]). $'i \in I : j \in N_i^{'}$ denotes the set of all demand points i which are in a neighborhood of candidate center j.

For an *M/G/1* queueing system, the waiting time is given by the Pollaczek-Khintchine formula (Kleinrock [24]):

$$W_j = \begin{cases} \dfrac{\lambda_j \overline{S_j^2}/2}{1-\lambda_j \overline{S_j}}, & \text{if } 1-\lambda_j\overline{S_j} > 0, \\ +\infty, & \text{otherwise,} \end{cases} \tag{2.7}$$

where $\overline{S_j}$ and $\overline{S_j^2}$ are the first and the second moments of the service time at center j, respectively. Hence, if the stability condition $(1-\lambda_j\overline{S_j} > 0)$ holds, then constraints (2.5) can be written as

$$\lambda_j\left(\frac{1}{2}\overline{S_j^2} + \tau_j\overline{S_j}\right) \le \tau_j\ . \tag{2.8}$$

Now, if we define $\overline{T_{ij}}$ and $\overline{T_{ij}^2}$ as the first and second moments of service times for the customer i at center j, then we have (see, e.g., Jamil et al. [17])

$$\begin{aligned} \overline{S_j} &= \sum_{i\in I:j\in N_i} h_i \overline{T_{ij}} x_{ij}, \\ \overline{S_{ij}^2} &= \sum_{i\in I:j\in N_i} h_i \overline{T_{ij}^2} x_{ij}, \end{aligned} \tag{2.9}$$

where h_i is the fraction of calls originating from demand point i which is defined as $h_i = f_i / \sum_{t\in I} f_t$. Then,

$$\begin{aligned} (2.8) \Longrightarrow & \left(\sum_{t\in I:j\in N_t} f_t x_{tj}\right)\left(\frac{1}{2}\sum_{i\in I:j\in N_i} h_i\overline{T_{ij}^2}x_{ij} + \tau_j \sum_{i\in I:j\in N_i} h_i\overline{T_{ij}}x_{ij}\right) \le \tau_j \\ \Longrightarrow & \left(\sum_{t\in I:j\in N_t} f_t x_{tj}\right)\left(\sum_{i\in I:j\in N_i} h_i\left(\frac{1}{2}\overline{T_{ij}^2} + \tau_j\overline{T_{ij}}\right)x_{ij}\right) \le \tau_j \\ \Longrightarrow & \sum_{i\in I:j\in N_i}\ \sum_{t\in I:j\in N_t}\left[f_t h_i\left(\frac{1}{2}\overline{T_{ij}^2} + \tau_j\overline{T_{ij}}\right)x_{tj}x_{ij}\right] \le \tau_j. \end{aligned} \tag{2.10}$$

Therefore, the problem with an *M/G/1* queue can be written as

$$\begin{aligned} \text{Max} \quad & \sum_{i\in I}\sum_{j\in N_i} a_i x_{ij}, \\ \text{s.t.} \quad & (2.2)\text{–}(2.4),\ (2.10) \text{ and } (2.6). \end{aligned} \tag{2.11}$$

To determine $\overline{T_{ij}}$ and $\overline{T_{ij}^2}$, we use the same assumptions as in Berman et al. [10]; that is, the average service time for demand point i from service center j consists of travel time to the scene, on the scene service time, travel time back to the center j, and possibly additional off-scene time. $\overline{T_{ij}}$ and $\overline{T_{ij}^2}$ are defined as

$$\begin{aligned}\overline{T_{ij}} &= \frac{\beta d(i,j)}{v} + \overline{Z_i},\\ \overline{T_{ij}^2} &= \left(\frac{\beta d(i,j)}{v}\right)^2 + 2\left(\frac{\beta d(i,j)}{v}\right)\left(\overline{Z_i}\right) + \overline{Z_i^2},\end{aligned} \tag{2.12}$$

where the parameter β is greater than or equal to one, $d(i,j)$ is the distance between demand point i and center at j, v is the speed, and $\overline{Z_i}$ is the average on-scene plus off-scene ("non-travel related") service time associated with node i.

This problem is NP-hard, and, in general, as the problem is a nonconvex programming problem, even small instances cannot be solved with the existing commercial softwares. A semidefinite programming relaxation for the problem is recently presented by Kakhki and Moghadas [25]. In Section 3 we discuss an approach for decomposing the problem into smaller subproblems and present some solution procedures.

3. A Solution Approach for Problem *P*

The basic idea for our proposed algorithm is to decompose the problem into smaller subproblems which is inspired by the solution methods suggested in the literature for the capacitated p-median problem (see, e.g., Baldacci et al. [26] and Lorena and Senne [27]).

Suppose that we have determined the location of $k-1(k \le p)$ centers and want to determine the location of the kth center from among the $n-k+1$ remaining candidate centers. If we remove the demand points assigned to the previous $k-1$ centers, then constraints (2.2) would be satisfied for any customer. In addition we consider the constraints (2.3) and (2.4) implicitly. Therefore, the problem can be decomposed into smaller binary quadratic subproblems (BQSP) as follows:

$$\begin{aligned}v_j &= \text{Max} \sum_{i\in\overline{I}:j\in N_i} a_i x_{ij},\\ \text{s.t.} &\sum_{i\in\overline{I}:j\in N_i}\ \sum_{t\in\overline{I}:j\in N_t} \left[f_t h_i \left(\frac{1}{2}\overline{T_{ij}^2} + \tau_j \overline{T_{ij}}\right) x_{tj} x_{ij}\right] \le \tau_j, \quad x_{ij} \in \{0,1\},\ \forall i \in \overline{I}, j \in J,\end{aligned} \tag{P_j}$$

where $\overline{I}$ is the set of remaining demand points, not assigned to the $k-1$ previous centers.

Now starting with $k=1$, we can solve $n-k+1$ binary quadratic subproblems P_j for all j not in the set of selected centers, J^*. We then find the optimal solution, j^*, delete all the demand points assigned to j^* from $\overline{I}$, and add j^* to J^*. The procedure is continued until $k \ge p$, or all the nodes are covered ($\overline{I} = \phi$). This procedure is outlined in Algorithm 1, below.

Solve $(P_j, \overline{I}\ )$ solves the binary quadratic subproblem P_j for a candidate center j. If a demand point $i(i \in \overline{I})$ is assigned to j, then i is added to I_j^*. *Update* $(\overline{I})$ removes all demand points assigned to j^* from $\overline{I}$.

This procedure can be easily extended to solve the more general case of fixed charge facilities; that is, 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

```
algorithm Heuristic procedure
begin
      Set $k = 1$,  $J^* = \phi$,  $\bar{I} = I$;
      While $k \le p$ & $\bar{I} \ne \phi$ do
      begin
         For all $j \notin J^*$ do
             $(v_j, I_j^*) = Solve\ (P_j, \bar{I})$;
         end;
         $j^* = \text{Arg}\ (\text{Max}\ \{v_j : j \notin J^*\})$;
         Let $J^* = J^* \cup \{j^*\}$ and $Update\ (\bar{I})$;
         $k = k + 1$;
      end;
end;
```

Algorithm 1: Heuristic algorithm for solving P.

the cost of establishing a center at site j. Notice that problem P is nonlinear and can be decomposed into smaller binary quadratic subproblems. Treating the quadratic terms $x_{ij}x_{tj}$'s as a single variable and using linearization methods, such as that of Sherali and Adams [28] will considerably increase the dimension of the problem. Instead we propose to solve the subproblem P_j using GRASP. In the next section, we discuss this procedure in more detail.

4. Solving Subproblem P_j with GRASP

GRASP (Greedy Randomized Adaptive Search Procedure) has been developed in late 1980s by Feo and Resende [29, 30]. Since then, it has been successfully applied to many problems, including the maximal covering location problem (Resende [31]). A bibliography of different applications is compiled by Festa and Resende [32, 33]. GRASP consists of two main phases: construction and local search. In the construction phase, a solution is built using a greedy function and randomization. In the local search, an optimal solution in the neighborhood of the solution found in the construction phase is obtained. Here we describe the details of the procedure for solving subproblems P_j with GRASP. Algorithm 2 shows the GRASP algorithm for solving BQSP P_j.

Algorithm 3 shows the details of the *preprocess* and *CGRS-LS* procedures. Here $W(I_j^*)$ is defined as $W(I_j^*) = \sum_{i \in I_j^*} a_i$. The procedure starts with $k = 1$, $key = 1$. The best solution is stored in $BestSolution$. The steps of the algorithm are repeated for $Maxitr$ iterations. If a new solution is constructed in the *construction phase*, then key is set equal to one, the *local search* is performed, and the best solution is updated.

The *restricted candidate list* (RCL) for our problem includes all demand points that can improve the objective while maintaining the waiting time constraint. If this list is empty, then key is set equal to zero and no local search is performed; otherwise, a candidate point is randomly selected and key is set to one, the local search is performed, and the best solution is updated.

Algorithm 4 shows the procedure for constructing the restricted candidate list. In this procedure, the so-called *candidate parameter* α ranges from zero to one. $\alpha = 0$ indicates that the

```
algorithm $GRASP - BQSP(\overline{I}, j)$
begin
    $preprocess$;
    While $k \le \mathrm{Max}\,i\,\mathrm{tr}$ & $key == 1$ do
    begin
        $CGRS - LS(\overline{I}, I_j^*, BestSolution)$;
        $k = k + 1$;
    end;
    $I_j^* = BestSolution, v_j = W(BestSolution)$;
    return $I_j^*, v_j$;
end;
```

Algorithm 2: *GRASP* procedure for solving BQSP P_j.

```
procedure $preprocess$
begin
  Set $k = 1, key = 1, I_j^* = \phi$;
  $BestSolution = \phi, W(BestSolution) = 0$;
end;


procedure $CGRS - LS(\overline{I}, I_j^*, BestSolution)$
begin
  $ConstructGreedyRandomSolution\_BQSP(\overline{I}, I_j^*)$;
  If $key == 1$ then
    $LocalSearch\_BQSP(I_j^*)$;
    If $W(I_j^*) > W(BestSolution)$ then
      $BestSolution = I_j^*$;
    end;
  end;
  return $\overline{I}, I_j^*, BestSolution$;
end;
```

Algorithm 3: *Preprocess* and *CGRS-LS* procedures.

points are randomly selected, while $\alpha = 1$ yields the greedy selection. $S1$ and R_{itj} are defined as

$$\begin{aligned} S1 &= \sum_{i \in I_j^*} \sum_{t \in I_j^*} f_i h_t \left(\frac{1}{2} \overline{T_{\imath\jmath}^2} + \tau_j \overline{T_{\imath\jmath}} \right), \\ R_{itj} &= f_i h_t \left(\frac{1}{2} \overline{T_{\imath\jmath}^2} + \tau_j \overline{T_{\imath\jmath}} \right). \end{aligned} \tag{4.1}$$

```
procedure $MakeRCL\_BQSP(\bar{I}, I_j^*)$
begin
  Set $RCL = \phi$;
  $a_{\max} = \max\{a_i : i \in \bar{I} \setminus I_j^*,\ S1 + R_{iij} + \sum_{t \in I_j^*}(R_{itj} + R_{tij}) \le \tau_j,$
                                 $stabilty\ condition\ is\ maintained\}$;
  For $i \in \bar{I} \setminus I_j^*$ do
      If $a_i \ge \alpha \times a_{\max}$ & $S1 + R_{iij} + \sum_{t \in I_j^*}(R_{itj} + R_{tij}) \le \tau_j$ then
        $RCL = RCL \cup \{i\}$
      end;
  end;
  return $RCL$;
end;
```

Algorithm 4: *MakeRCL_BQSP* procedure.

In this algorithm $S1$ is actually the left hand side of (2.10). If demand point i $(i \in \bar{I} \setminus I_j^*)$ is added to I_j^*, then the left hand side of this equation is increased by $R_{iij} + \sum_{t \in I_j^*}(R_{itj} + R_{tij})$, so $S1 + R_{iij} + \sum_{t \in I_j^*}(R_{itj} + R_{tij})$ must be less than τ_j.

In the *local search*, we use the 2-exchange neighborhood structure. The exchange between two demand points $i \in I_j^*$ and $s \in \bar{I} \backslash I_j^*$ is only possible if $a_s > a_i$, and, in addition, adding s to I_j^* and deleting i from I_j^* would not violate the constraints. The procedure for local search is shown in Algorithm 5.

If $s \in \bar{I} \backslash I_j^*$ is added to and $i \in I_j^*$ is deleted from I_j^*, then $S1$ is increased by $R_{ssj} + \sum_{t \in I_j^*}(R_{tsj} + R_{stj})$ and decreased by $R_{iij} + \sum_{t \in I_j^*,\, t \neq i}(R_{itj} + R_{tij})$. In the above procedure, $i \in I_j^*$ is replaced by a demand point r $(r \in \bar{I} \backslash I_j^*)$ which has maximum population and would not violate the constraints.

Finally, in construction and local search phases, adding $i \in \bar{I}$ to I_j^* is possible only if the stability condition is not violated.

5. GRASP for Problem *P*

Our procedure for solving P is similar to that proposed for the maximal covering problem by Resende [31]). The main difference is in selection of the greedy function for construction of the restricted candidate list. The construction of the restricted candidate list is illustrated in Algorithm 6.

We use the GRASP procedure to solve the subproblem P_j for any candidate facility j $(j \in J \setminus J^*)$, as indicated in line 5.

There are few differences between our implementation of GRASP and that of Silva and Serra [18] for the *M/M/1* case with priorities. In Silva and Serra [18], the allocation is based on a set D_{ij} which is constructed for each candidate center j. D_{ij} contains the indices of all demand points i for which $d(i, j)$ is less than R. Entries in D_{ij} are ordered according to their distances from j. Demand points from D_{ij} are assigned to j as long as the waiting time constraint is not violated. Another difference is in the selection of the greedy function.

```
procedure LocalSearch_BQSP ($I_j^*$)
begin
    Set $key = 1$;
    While $key == 1$ do
    begin
        $key = 0$;
        For all $i \in I_j^*$ do
          $s_{\max} = i$ , $a_{\max} = a_i$ ;
          For all $s \in \bar{I} \setminus I_j^*$ do
              If $(S1 - (R_{iij} + \sum_{t \in I_j^*, t \neq i}(R_{itj} + R_{tij})) + (R_{ssj} + \sum_{t \in I_j^*}(R_{tsj} + R_{stj})) \leq \tau_j)$ & $(a_s > a_i)$
                  & stability condition is maintained then
              If $a_s > a_{\max}$ then
                    $a_{\max} = a_s$ , $s_{\max} = s$;
              end;
            end;
          end;
          If $s_{\max} \neq i$ then
            $r = s_{\max}$ , $key = 1$;
            $S1 = S1 - (R_{iij} + \sum_{t \in I_j^*, t \neq i}(R_{itj} + R_{tij})) + (R_{rrj} + \sum_{t \in I_j^*}(R_{trj} + R_{rtj}))$;
            $I_j^* = I_j^* \cup \{r\} \setminus \{i\}$ , $\bar{I} = \bar{I} \cup \{i\} \setminus \{r\}$;
          end;
        end;
    end;
    return $I_j^*$;
end;
```

Algorithm 5: *LocalSearch_BQSP* procedure.

```
procedure MackRCL–MainProblem($J^*$, $\bar{I}$)
begin
      set $RCL\_MP = \phi$;
      For all $j \in J \setminus J^*$ do
          $[I_j^*, v_j] \leftarrow GRASP\_BQSP(\bar{I}, j)$;
      end;
      $v_{\max} = \max\{v_j : j \in J \setminus J^*\}$;
      $RCL\_MP = \{j \mid j \in J \setminus J^*, v_j \geq \alpha \times v_{\max}\}$;
      return $RCL\_MP$;
end;
```

Algorithm 6: *MackRCL–MainProblem* procedure.

In Silva and Serra [18], the greedy function is the total costumer arrival rate, while ours is the amount of increase in the objective function. Finally their local search is a comprehensive search over all feasible solutions which improves the objective, while we use a 2-exchange neighborhood search.

Table 1: Computational results for test problems solved with the heuristic procedure.

Number of demand points	Percent of population covered					Least number of facilities needed for 100% pop. coverage	Average CPU time for 100% pop. coverage (seconds)
	$p = 1$	$p = 3$	$p = 5$	$p = 10$	$p = 15$		
20	77.05	98.10	—	—	—	4	0.79
30	70.74	95.24	100	—	—	5	2.30
50	47.33	72.57	85.52	—	—	9	6.58
100	34.22	58.90	74.22	94.18	100	15	55.97
150	26.23	48.34	66.34	89.77	98.46	18	192.85
200	22.37	43.46	60.64	86.36	95.74	22	468.40

6. Computational Results

The queueing maximal covering location-allocation problem with an *M/G/1* is considerably more difficult to solve than the one with an *M/M/1* system, since the waiting time (quality of service) constraints are quadratic with possibly indefinite coefficient matrices. Hence, even small-sized problems cannot be solved with the state-of-the-art software packages of today. Here we have tried to solve some randomly generated test problems by the proposed algorithms. We first considered the solution of problems with heuristic algorithm where subproblems P_j were solved with GRASP. Then we used the GRASP to solve the entire model P. All calculations were performed on a Pentium IV processor with 2.80 GHz and 2.50 GB of RAM.

The number of points in the instances ranged from 20 to 200. In these instances, 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, and the distances are considered to be Euclidean.

f_i and τ_j, namely, the daily call rate and the average time limit, were taken to be 0.005 times the population and 12.75 minutes for each candidate center. The covering radius, R, was taken to be 1.5 miles. Parameters α and β were set to 0.85 and 2; respectively, and v was set to 5 (miles/hour). $\overline{Z_{\iota}}$ is assumed to be the same for all demand points and was taken to be 0.5 (hour). The maximum number of iterations was set to 20. Each problem was run 5 times. Percent of coverage listed are the averages. Tables 1 and 2 show the results for instances solved with the suggested heuristic and with the GRASP procedure, respectively.

Comparison of the results for the two approaches reveals that solving the subproblems with GRASP takes far less time than does solving the entire model. The CPU time increases dramatically for the second algorithm as the number of points increases, while the coverage does not improve substantially. For example, as indicated in Tables 1 and 2, for the 200 point test problem, the first approach provides a solution with 100% coverage by selecting $p = 22$ in 468.40 seconds, while the second approach gives a solution with 100% coverage with $p = 21$ in 9905.76 seconds.

Finally, in order to have some idea about our solution methods, we tried to solve some small instances by enumerating all possible outcomes. The results for 5 and 10 points with 1 and 2 centers are shown in Table 3.

Table 2: Computational results for test problems solved with GRASP.

Number of demand points	Percent of population covered					Least number of facilities needed for 100% pop. coverage	Average CPU time for 100% pop. coverage (seconds)
	$p = 1$	$p = 3$	$p = 5$	$p = 10$	$p = 15$		
20	77.05	100	—	—	—	3	24.36
30	70.74	98.09	100	—	—	5	63.45
50	47.33	74.85	87.08	—	—	9	194.24
100	34.22	60.05	75.20	95.77	100	15	1270.66
150	26.23	51.37	68.13	91.16	98.79	18	4205.43
200	22.37	45.38	62.17	87.31	96.61	21	9905.76

Table 3: Comparison of the results for enumerative method with the heuristic procedure and GRASP.

		$m = n = 5, p = 1$	$m = n = 10, p = 1$	$m = n = 10, p = 2$
Enumeration method	Percent of population covered	100	90.47	100
	Population covered	2630	3230	3570
	Selected center(s)	1 or 2 or 4 or 5	2	2,5
Heuristic procedure	Percent of population covered	100	90.47	100
	Population covered	2630	3230	3570
	Selected center(s)	1 or 2 or 4 or 5	2	2,1
GRASP	Percent of population covered	100	90.47	100
	Population covered	2630	3230	3570
	Selected center(s)	1 or 2 or 4 or 5	2	5,10

For these limited instances, there were no discrepancies between the solutions. However, this of course is no proof that the algorithm always obtains an optimal solution, but we hope and expect it to at least obtain a rather good solution.

7. Conclusion

In this paper we considered the queueing maximal covering problem with an *M/G/1* queueing system. A quadratically constrained integer programming model was presented for the problem. To solve the problem, we first considered a heuristic algorithm based on decomposing the problem into smaller knapsack-type subproblems. Then we solved the binary quadratic subproblems with GRASP. We also used GRASP to solve the entire model. Our limited computational results showed that using the proposed heuristic algorithm provide better solutions.

Acknowledgment

The authors would like to express their gratitude to the anonymous referees for their helpful comments. This research was supported in part by a grant from Ferdowsi University of Mashhad, No. MA86247TAG.

References

[1] R. Church and C. ReVelle, “The maximal covering location problem,” *Papers of the Regional Science Association*, vol. 32, no. 1, pp. 101–118, 1974.

[2] C. ReVelle, “Review, extension and prediction in emergency service siting models,” *European Journal of Operational Research*, vol. 40, no. 1, pp. 58–69, 1989.

[3] D. A. Schilling, V. Jayaraman, and R. Barkhi, “A review of covering problems in facility location,” *Location Science*, vol. 1, no. 1, pp. 25–55, 1993.

[4] V. Marianov and C. ReVelle, “The queueing maximal availability location problem: a model for the siting of emergency vehicles,” *European Journal of Operational Research*, vol. 93, no. 1, pp. 110–120, 1996.

[5] L. Brotcorne, G. Laporte, and F. Semet, “Ambulance location and relocation models,” *European Journal of Operational Research*, vol. 147, no. 3, pp. 451–463, 2003.

[6] V. Marianov and D. Serra, “Location problems in the public sector,” in *Facility Location: Application and Theory*, Z. Drezner and H. W. Hamacher, Eds., pp. 119–150, Springer, 2004.

[7] J. B. Goldberg, “Operations research models for the deployment of emergency services vehicles,” *EMS Management Journal*, vol. 1, no. 1, pp. 20–39, 2004.

[8] C. S. ReVelle, H. A. Eiselt, and M. S. Daskin, “A bibliography for some fundamental problem categories in discrete location science,” *European Journal of Operational Research*, vol. 184, no. 3, pp. 817–848, 2008.

[9] E. Erkut, A. Ingolfsson, T. Sim, and G. Erdoğan, “Computational comparison of five maximal covering models for locating ambulances,” *Geographical Analysis*, vol. 41, no. 1, pp. 43–65, 2009.

[10] O. Berman, R. C. Larson, and S. S. Chiu, “Optimal server location on a network operating as an M/G/1 queue,” *Operations Research*, vol. 33, no. 4, pp. 746–771, 1985.

[11] O. Berman and R. R. Mandowsky, “Location-allocation on congested networks,” *European Journal of Operational Research*, vol. 26, no. 2, pp. 238–250, 1986.

[12] O. Berman, R. C. Larson, and C. Parkan, “The stochastic queue p-median location problem,” *Transportation Science*, vol. 21, pp. 207–216, 1987.

[13] R. Batta and O. Berman, “A location model for a facility operating as an M/G/k queue,” *Networks*, vol. 19, pp. 717–728, 1989.

[14] R. Batta, “A queueing-location model with expected service time dependent queueing disciplines,” *European Journal of Operational Research*, vol. 39, no. 2, pp. 192–205, 1989.

[15] V. Marianov and C. Revelle, “The queuing probabilistic location set covering problem and some extensions,” *Socio-Economic Planning Sciences*, vol. 28, no. 3, pp. 167–178, 1994.

[16] V. Marianov and D. Serra, “Probabilistic, maximal covering location-allocation models from congested systems,” *Journal of Regional Science*, vol. 38, no. 3, pp. 401–424, 1998.

[17] M. Jamil, A. Baveja, and R. Batta, “The stochastic queue center problem,” *Computers and Operations Research*, vol. 26, no. 14, pp. 1423–1436, 1999.

[18] F. Silva and D. Serra, “Locating emergency services with different priorities: The priority queuing covering location problem,” *Journal of the Operational Research Society*, vol. 59, no. 9, pp. 1229–1238, 2008.

[19] F. M. Moghadas and H. T. Kakhki, “Maximal covering location-allocation problem with M/M/k queueing system and side constraint,” *Iranian Journal of Operations Research*, vol. 2, no. 2, pp. 1–16, 2011.

[20] O. Berman and D. Krass, “Facility location problems with stochastic demands and congestion,” in *Facility Location: Application and Theory*, Z. Drezner and H. W. Hamacher, Eds., pp. 329–371, Springer, 2004.

[21] F. D. A. Corrêa and L. A. N. Lorena, “Using the constructive genetic algorithm for solving the probabilistic maximal covering location-allocation problem,” in *Proceedings of the Workshop on Computational Intelligence*, SBRN, 2006.

[22] F. D. A. Corrêa, A. A. Chaves, and L. A. N. Lorena, “Hybrid heuristics for the probabilistic maximal covering location-allocation problem,” *Operational Research*, vol. 7, no. 3, pp. 323–344, 2008.

[23] F. A. Corrêa, L. A. N. Lorena, and G. M. Ribeiro, “A decomposition approach for the probabilistic maximal covering location-allocation problem,” *Computers and Operations Research*, vol. 36, no. 10, pp. 2729–2739, 2009.

[24] L. Kleinrock, *Queueing Systems: Theory*, vol. 1, John Wiley & Sons, 1975.

[25] H. T. Kakhki and F. M. Moghadas, “A semidefinite programming relaxation for the queueing covering location problem with an M/G/1 system,” in *Proceedings of the European Workshop on Mixed Integer Nonlinear Programming*, pp. 231–236, CIRM, April 2010.

[26] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, and A. Mingozzi, "A new method for solving capacitated location problems based on a set partitioning approach," *Computers and Operations Research*, vol. 29, no. 4, pp. 365–386, 2002.

[27] L. A. N. Lorena and E. L. F. Senne, "Local search heuristics for capacitated p-median problems," *Networks and Spatial Economics*, vol. 3, no. 4, pp. 407–419, 2003.

[28] H. D. Sherali and W. P. Adams, "A tight linearization and an algorithm for zero-one quadratic programming problems," *Management Science*, vol. 32, no. 10, pp. 1274–1290, 1986.

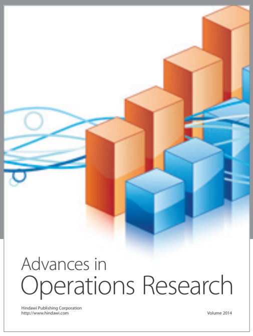
[29] T. A. Feo and M. G. C. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.

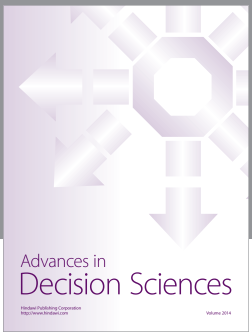
[30] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.

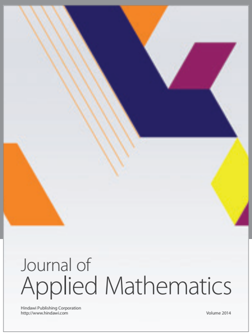
[31] M. G. C. Resende, "Computing approximate solutions of the maximum covering problem with GRASP," *Journal of Heuristics*, vol. 4, no. 2, pp. 161–177, 1998.

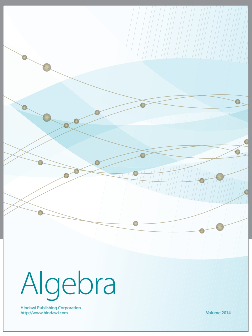
[32] P. Festa and M. G. C. Resende, "An annotated bibliography of GRASP—part I: algorithms," *International Transactions in Operational Research*, vol. 16, pp. 1–24, 2009.

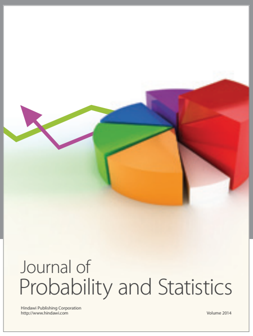
[33] P. Festa and M. G. C. Resende, "An annotated bibliography of GRASP—spart II: applications," *International Transactions in Operational Research*, vol. 16, pp. 131–172, 2009.


Advances in
Operations Research

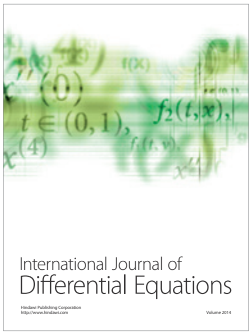

Advances in
Decision Sciences

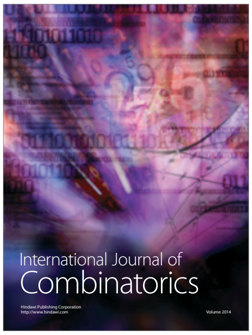

Journal of
Applied Mathematics

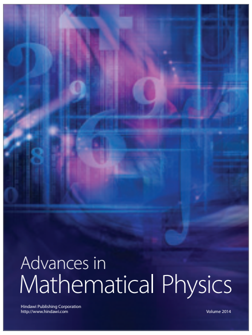

Algebra

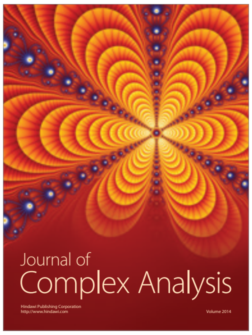

Journal of
Probability and Statistics

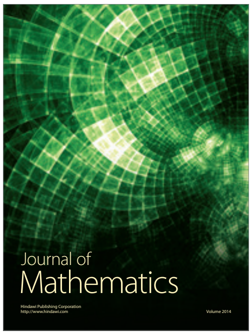
The Scientific
World Journal

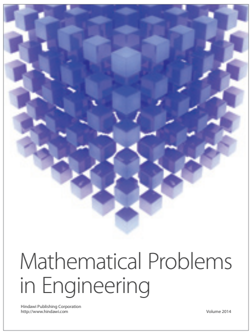

International Journal of
Differential Equations

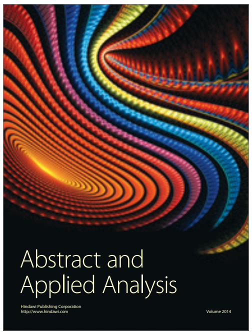

International Journal of
Combinatorics

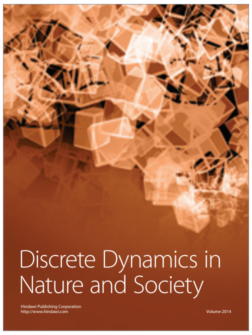

Advances in
Mathematical Physics

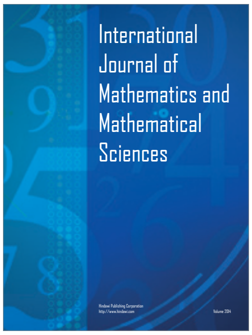

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

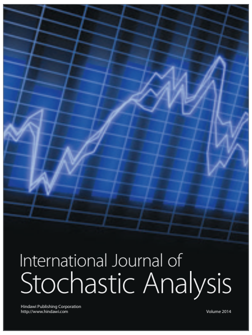

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization