Emergency Location Problems with an M/G/k Queueing System

F. Moeen-Moghadas*,1, E. Monabbati2, H. Taghizadeh-Kakhki3

Since late 1960's, the emergency location problems, fire stations and medical emergency services have attracted the attention of researchers. Mathematical models, both deterministic and probabilistic, have been proposed and applied to find suitable locations for such facilities in many urban and rural areas. Here, we review some models proposed for finding the location of such facilities, with an eye on successfully implemented real life applications. We then propose an extension of the QM-CLAM model of Marianov and Serra (1998) to M/G/k systems, and suggest a GRASP type heuristic procedure for solving the problem. To improve the computed solution, local search heuristics are used. Sensitivity analysis and some computational results are also presented.

Keywords: Emergency location problems, Maximal covering location problems, Queueing location models, GRASP.

1. Introduction

Emergency location problems deal with finding suitable locations for emergency facilities, such as police, fire, and ambulance stations. According to [26], one of the earliest such works was done for the New York city fire department by the RAND corporation. A paper by Savas [48] which presents a simulation analysis of an ambulance system for a hospital in New York along with the paper by Carter and Ignall [11] on simulation of fire fighting operations, were perhaps the first such attempts to aid the decision makers. One of the earliest optimization models presented for this problem is the Location Set Covering Model (LSCM) of Toregas et al. [56]. This simple model aims to find the least number of ambulances needed to cover all demand points.

Another important model called Maximal Covering Location Problem (MCLP) was presented by Church and ReVelle [13]. The objective in this model was to maximize the population covered by a given number of ambulances. This model was later used by Plane and Hendrick [42] to find the location of fire stations in Denver, Colorado. It was also used by Eaton et al. [18] to reorganize the medical emergency services in the city of Austin, Texas. Later, Eaton et al. [17] used a multi-objective formulation of the problem to find the locations of emergency service facilities for Santo Domingo in the Dominican Republic. A set covering approach was used by Schreuder [50] to calculate the minimum number of fire stations for the city of Rotterdam to reach any point in the city within a prescribed attendance time.

*Corresponding Author.
1 Department of Mathematics, University of Bojnord, Bojnord, Iran. E-mail: forooghmoene@gmail.com
2 Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran. E-mail: eh_mo236@stu-mail.um.ac.ir
3 Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran. E-mail: taghizadi@um.ac.ir
Neither the LSCM nor the MCLP model considers the type of ambulances used. A model which took into account different types of emergency vehicles is the Tandem Equipment Allocation Model (TEAM) of Schilling et al. [49]. A variant of this model known as FLEET (Facility Location and Equipment Emplacement Technique) was used to locate fire stations and depots in the city of Baltimore, Maryland.

Along with these deterministic models, many probabilistic ones have also been proposed. One of the earliest such models is the probabilistic set covering model of Chapman and White [12]. Another model is the Maximum Expected Covering Location Problem (MEXCLP) of Daskin [16]. The objective in this model was to maximize the total expected demand covered. Fujiwara et al. [23] used this model to find the emergency centers for the city of Bangkok.

The Queueing Probabilistic Location Set Covering Problem (QPLSCP) was proposed by Marianov and ReVelle [33] in which the busy fraction for each server was considered to be different. They later proposed the QMALP model with an M/G/s-loss queueing system [34].

Another extension of the MCLP model, called QM-CLAP, was proposed by Marianov and Serra [35] with additional constraints to model congestion. Assuming that the demand has a Poisson distribution and the service time is exponential, they formulate the problem as M/M/1 and M/M/m queueing systems.

Determining the location of a center when the queueing system is M/G/1 was discussed by Jamil et al. [29]. Batta [5] also presented an M/G/1 model, SQCL, in which a weighted combination of the average and the variance of waiting time is minimized. Another queueing model with priority was proposed by Silva and Serra [53].

More discussion on most of the aforementioned models can be found in the review papers of ReVelle [46], Marianov and ReVelle [32], and Brotcorne et al. [10]. A recent paper of Li et al. [31] considers several covering models and solution algorithms for Emergency Medical Service (EMS) allocation problems. A bibliography of maximal covering problems is also given in the review paper of ReVelle et al. [47]. A review of the location problems with stochastic demands and constraints on waiting times can be found in Baron et al. [4].

In addition to the mentioned covering models, other models of the P-center and P-median types have also been proposed to locate emergency centers. Garfinkel et al. [24] were perhaps one of the earliest to use a P-center model to locate emergency facilities on a road network. Hochbaum and Pathria [28] formulated the problem of finding emergency centers as a probabilistic P-center problem. Talmar [55] formulated and solved a real life problem of locating and dispatching three helicopters in the Alpine mountain ranges to respond to ski and climbing accidents in that area, as a P-center problem.

One of the early P-median models proposed for the emergency centers is the model proposed by Mirchandani [37] for finding the location of fire stations, in which the demand and travel times are assumed to be probabilistic. Another model that tries to minimize the average cost of response is the Stochastic Queue Median (SQM) model of Berman et al. [7]. Carson and Batta [10] present a dynamic P-median model for locating ambulances in a university campus.
Table 1. Some real world applications of EMS models

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Location</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEXLCP</td>
<td>1974</td>
<td>Tucson, Arizona</td>
<td>[41]</td>
</tr>
<tr>
<td>MCLP</td>
<td>1977</td>
<td>Denver, Colorado</td>
<td>[42]</td>
</tr>
<tr>
<td>FLEET</td>
<td>1979</td>
<td>Baltimore, Maryland</td>
<td>[49]</td>
</tr>
<tr>
<td>Set covering</td>
<td>1981</td>
<td>Rotterdam, Netherlands</td>
<td>[50]</td>
</tr>
<tr>
<td>MCLP</td>
<td>1985</td>
<td>Austin, Texas</td>
<td>[17]</td>
</tr>
<tr>
<td>MCLP</td>
<td>1986</td>
<td>Santo Domingo</td>
<td>[17]</td>
</tr>
<tr>
<td>MEXLCP</td>
<td>1989</td>
<td>Bangkok, Thailand</td>
<td>[23]</td>
</tr>
<tr>
<td>AMEXCLP</td>
<td>1990</td>
<td>Tucson, Arizona</td>
<td>[25]</td>
</tr>
<tr>
<td>P-median</td>
<td>1990</td>
<td>Amherst, Buffalo</td>
<td>[10]</td>
</tr>
<tr>
<td>TIMEXCLP</td>
<td>1994</td>
<td>Louisville, Kentucky</td>
<td>[44]</td>
</tr>
<tr>
<td>P-median</td>
<td>1998</td>
<td>Barcelona, Spain</td>
<td>[51]</td>
</tr>
<tr>
<td>Multi-criteria</td>
<td>1998</td>
<td>Dubai, UAE</td>
<td>[3]</td>
</tr>
<tr>
<td>P-center</td>
<td>2002</td>
<td>Alpine mountain</td>
<td>[55]</td>
</tr>
<tr>
<td>P-median</td>
<td>2002</td>
<td>Barbados</td>
<td>[27]</td>
</tr>
<tr>
<td>P-median</td>
<td>2004</td>
<td>Carbondale, Illinois</td>
<td>[41]</td>
</tr>
<tr>
<td>P-median</td>
<td>2006</td>
<td>Riyadh, Saudi Arabia</td>
<td>[1]</td>
</tr>
<tr>
<td>Probabilistic MCLP</td>
<td>2009</td>
<td>Edmonton, Canada</td>
<td>[19]</td>
</tr>
<tr>
<td>MEXCLP2</td>
<td>2009</td>
<td>Hanover, Virginia</td>
<td>[36]</td>
</tr>
<tr>
<td>MERLP</td>
<td>2009</td>
<td>Charlotte, North Carolina</td>
<td>[43]</td>
</tr>
<tr>
<td>A Covering model</td>
<td>2010</td>
<td>Adana, Turkey</td>
<td>[15]</td>
</tr>
<tr>
<td>Multi-period backup</td>
<td>2011</td>
<td>Istanbul, Turkey</td>
<td>[2]</td>
</tr>
<tr>
<td>Set covering</td>
<td>2012</td>
<td>Tehran, Iran</td>
<td>[52]</td>
</tr>
</tbody>
</table>

A bi-objective model was proposed by Harewood [27] to locate ambulances on the island of Barbados. A bi-objective formulation of a covering problem as a goal programming problem was presented by Alsalloum and Rand [1]. This model was used to determine the location of EMS centers for the Red Cross in the city of Riyadh. Badri et al. [3] presented a multi-criteria model for locating fire stations in the city of Dubai, UAE. We have compiled some real life applications of some of these models, in chronological order, as shown in Table 1, to emphasize the importance and applicability of these mathematical models.

It is argued by some researchers (see e.g., Borras and Pastor [8]), that probabilistic models provide a more accurate presentation of the real world problem. In a study by Erkut et al. [19], four probabilistic variants of the maximal covering location problem were compared with the basic model, using data from Edmonton, Canada. They showed that a model that incorporates uncertainty covers up to 26% more of the demand. Recently, Moeen Moghadas and Taghizadeh [38] considered the problem with an M/M/k system with side constraints on the number of servers at each center, as well as constraints on the total cost of establishing centers.

An extension of the QM-CLAP model of Marianov and Serra [35] with an M/G/1 queueing system was proposed by Moeen-Moghadas and Taghizadeh [39]. They also discussed a semi-definite programming relaxation of a binary quadratic formulation of the M/G/1 model [54]. Here, we extend the results of [39] to the M/G/k queueing systems, where $k$ itself is unknown, but there is a limit on the number of servers at each center, and the total number of servers in the system. We propose a
solution procedure, and present numerical examples to illustrate the applicability of the proposed model.

2. Mathematical Model

The QM-CLAP model proposed by Marianov and Serra [35] is as follows:

\[
\begin{align*}
\text{max} & \quad \sum_{i \in I} \sum_{j \in N_i} a_i x_{ij} \\
\text{s.t.} & \quad x_{ij} \leq y_j, \quad i \in I, j \in N_i \quad (1) \\
& \quad \sum_{j \in N_i} x_{ij} \leq 1, \quad i \in I \quad (2) \\
& \quad \sum_{j \in j} y_j \leq p, \quad (3) \\
& \quad W_j \leq \tau, \quad j \in J \quad (4) \\
& \quad x_{ij}, y_j \in \{0, 1\}, \quad i \in I, j \in N_i. \quad (5)
\end{align*}
\]

where,

\( I \): The set of all existing demand points (incident locations),

\( J \): The set of all possible locations of new facilities (stations),

\( N_i \): The set of all centers in a predetermined neighborhood of point \( i \); i.e.,

\[ N_i = \{ j : d_{ij} \leq D, \text{for a given distance } D \} \]

Where, \( d_{ij} \) is the distance between customer \( i \) and center \( j \), \( y_j \), equals 1, if a new facility is located at site \( j \), and 0, otherwise,

\( x_{ij} \): Equals 1, if a call from point \( i \) is answered by station \( j \), and 0, otherwise,

\( a_i \): Population at point \( i \),

\( p \): The number of new stations,

\( W_j \): Waiting time at facility \( j \),

\( \tau \): Maximum allowable waiting time.

The objective maximizes the population covered. Constraints (1) ensure that a point is being served only by an established facility at \( i \). Constraints (2) guarantee that each point is being served only by one service facility (station). Constraints (3) establish at most \( p \) new stations, and constraints (4) ensure that the waiting time at each station does not exceed a predetermined amount.

This model is a modification of the well known P-median problem, with the constraint set (4), quality of service constraints, added. To state the problem properly, we need an explicit form expressing the waiting time, \( W_j \), in terms of the decision variables \( x_{ij} \) and \( y_j \). This, in turn, is system dependent. For the M/M/1 and M/M/m systems, Marianov and Serra [35] presented a linear form. They assume that arriving calls have the intensity \( f_i \), and hence, the arrival rate at each center \( j \), \( \lambda_j \), could be calculated as \( \lambda_j = \sum_i f_i x_{ij} \). Thus, constraints (4) can be replaced by

\[
\sum_{i \in I, j \in N_i} f_i x_{ij} \leq \frac{\mu^2 \tau}{1 + \mu \tau}.
\]
For an M/G/1 queueing system, however, unfortunately we do not have such a linear representation, as the waiting time is given by (See e.g., [30]):

\[ W_j = \frac{\lambda_j \bar{S}_j^2}{2(1 - \lambda_j \bar{S}_j)}, \quad 1 - \lambda_j \bar{S}_j > 0, \]

where \( \bar{S}_j = E[S_j] \) and \( \bar{S}_j^2 = E[S_j^2] \) are respectively the first and the second moments of the service time at center \( j \). It can be shown that under certain assumptions, this leads to a quadratic constraint with possibly an indefinite coefficient matrix; hence, a non-convex feasible region. As a result, the model cannot be solved easily, even for small instances.

In our M/G/k model, we assume that there are a total of \( K \) servers available and that each center can have at most \( k_{\text{max}} \) servers. In addition, if we define the variable \( k_j \) as the number of servers at the center \( j \), \( (k_j = 0, 1, \cdots, k_{\text{max}}) \), then the problem can be stated as follows:

\[
\begin{align*}
\max & \quad \sum_{i \in I} \sum_{j \in \mathcal{N}_i} a_{ij} x_{ij} \\
\text{s.t.} & \quad (1) \text{ through } (5), \\
& \quad \sum_{j \in \mathcal{S}} k_j \leq K, \\
& \quad k_j \leq k_{\text{max}} y_j, \quad j \in \mathcal{S}, \\
& \quad k_j = 0, 1, \cdots, k_{\text{max}}, \quad j \in \mathcal{S}.
\end{align*}
\]

In this model, the objective function and constraints (1)-(5) are defined as before. Constraint (6) limits the number of servers at all selected centers and constraints (7) ensure that at most \( k_{\text{max}} \) servers are placed at each center.

To state the problem properly, however, we need an explicit form expressing the average waiting time in terms of the variables \( x_{ij}, y_j \) and \( k_j \). Unfortunately, in the case of M/G/k queueing system, there is not an explicit form for the average waiting time. We need to use an approximate formula for the waiting time. Here, we opt for the approximation proposed by Nozaki and Ross [40], which is shown to be “adequate” by Batta and Berman [6]. This approximating formula is:

\[
W(\lambda_j, k_j) = \begin{cases} \\
\lambda_j^{k_j-1} \frac{S_j^{k_j}}{1/(1-\lambda_j S_j)} - 1 \frac{2(k_j - 1)!(k_j - \lambda_j S_j)}{S_j} \sum_{t=0}^{k_j-1} \frac{(k_j - t)(\lambda_j S_j)^t}{t!}, & \lambda_j S_j < k_j \\
+\infty, & \text{otherwise,}
\end{cases}
\]

where \( W(\lambda_j, k_j) \) is the average waiting time at center \( j \) with arrival rate \( \lambda_j \) and with \( k_j \) servers, \( \bar{S}_j \) and \( \bar{S}_j^2 \) are as defined before. Now, if we define \( \bar{T}_{ij} \) and \( \bar{T}_{ij}^2 \) as the first and second moments of service time for the customer \( i \) at center \( j \), respectively, then we have [29]:

\[
\bar{S}_j = \sum_{i \in I: j \in \mathcal{N}_i} h_i \bar{T}_{ij} x_{ij},
\]
\[ S_j^2 = \sum_{i \in l, j \in N_i} h_i \bar{T}_{ij}^2 x_{ij}, \]  
\hfill \text{(10)}

where \( h_i \) is the fraction of calls originating from demand point \( i \) which is defined to be \( h_i = \frac{f_i}{\sum_i f_i} \).

To determine \( \bar{T}_{ij} \) and \( \bar{T}_{ij}^2 \), we use the same assumptions as in Berman et al. [7]; i.e., we assume the average service time at point \( i \) served by center \( j \) consists of four parts: travel time to the scene, on the scene service time, travel time back to the center \( j \), and possibly additional off scene time. If we define \( v \) as the travel speed, then \( \bar{T}_{ij} \) and \( \bar{T}_{ij}^2 \) can be written as [7]:

\[ \bar{T}_{ij} = \frac{\beta d_{ij}}{v} + Z_i, \]
\[ \bar{T}_{ij}^2 = \left( \frac{\beta d_{ij}}{v} \right)^2 + 2 \frac{\beta d_{ij}}{v} + Z_i^2, \]

where \( Z_i \) is the average on scene plus off scene (non-travel related) service time associated with node \( i \) and \( \beta \geq 1 \) is a parameter.

Note that if we replace (9) and (10) in the approximation formula, the problem will be a nonlinear integer programming problem which is NP-hard; hence, even small problems cannot be solved by the existing commercial software. In the next section, we propose a heuristic solution procedure for this problem.

3. Solution Procedure

For the M/M/1 and M/M/m cases, as mentioned, we have an integer linear program, which can be solved by commercial software packages for small problems. For larger problems, heuristic methods have been developed by Marianov and Serra [35].

A GRASP type procedure for the priority queueing location problem has also been proposed by Silva and Serra [53]. Another approach for solving QM-CLAP is the procedure proposed by Correa et al. [14]. The authors in [14] model the problem as a covering graph. To the best of our knowledge, no study has considered this model when the underlying system is an M/G/k queue. Here, we propose a GRASP type procedure for solving the problem. We complement this procedure with two local search heuristics to improve the solution. Greedy Randomized Adaptive Search Procedure (GRASP) was developed by Feo and Resende [20], and successfully applied to many problems, including maximal covering problem [45]. A two-part annotated bibliography of GRASP was presented by Festa and Resende in [21] and [22]. Each GRASP iteration consists of constructing a solution and then performing a local search around the constructed solution.

Following the developments in [39], if we relax constraints (1)–(3), then we will have knapsack sub-problems of the following form:

\[ v(P_j) = \max_{(P_j)} \sum_{i \in l} \sum_{j \in N_i} a_i x_{ij} \]
\[ \text{s.t.} \quad W(\lambda_j, k_j) \leq \tau, \quad j \in J \]
\[ x_{ij} \in \{0,1\}, \quad i \in l, j \in N_i. \]
To solve the sub-problems, we use a modified version of the heuristic procedure GRASP.

The GRASP procedure that we use is in principle similar to the one proposed for the maximal covering problem by Resende [45]. The main differences are in construction of the restricted candidate list and the greedy function. In the construction phase (subprogram CGRS given as Fig. 1), an initial feasible set of solutions and an assignment of “demand” points to these centers are constructed; a set RCL of candidate centers which can improve the objective function is determined (Fig. 2). Then, $k$ centers are randomly selected from this set. If $RCL \neq \emptyset$, (key=1), then the sets $J^*$ and $\bar{I}$, the set of selected centers and the set of “demand” points not yet received a service, respectively, are updated. In doing so, we will take constraints (1) and (2) into account. $k$, $k_j$ and $k_{max}$, are the number of centers selected, number of servers at center $j$, and the total number of servers, which should not exceed a maximum number, $\bar{k}$. In this phase, finding a new center is continued until all centers are located, or the number of servers exceed $\bar{k}$, or the set of candidate centers is empty.

To construct the set RCL, sub-problem $P_j$ is solved for all $j \not\in J^*$. Then, all the centers with an objective value of at least $\alpha$ ($0 < \alpha < 1$) times the maximum value of the objectives for $j \not\in J^*$ are selected. $\alpha = 0$ means that the centers are selected randomly, while $\alpha = 1$ indicates that the centers are selected using a greedy procedure.

In the local search phase, a 2-exchange procedure is employed.

Note that in sub-problem $P_j$, $k_j$ and $x_{ij}$ are not known. Because the set of demand points assigned to the center $j$ ($I'_j$) are dependent on the number of servers at $j$ ($k_j$), we solve this problem for different

$$k_{sum} = 0, k = 0, key = 1$$

while (( $k < p$) and ($k_{sum} < \bar{k}$) and ($key == 1$)) do
    RCL = MakeRCL($J^*, \bar{I}$)
    if (RCL $\neq \emptyset$) then
        $s = SelectFacility(RCL)$
        Update($J^*, \bar{I}$)
        $k_{sum} = k_{sum} + k_s$
        $k = k + 1$
    else
        $key = 0$
    end
end
return $J^*$

Figure 1. The construction procedure CGRS() for GRASP

for ($j \not\in J^*$) do
    $v(P_j) = SolveSubProblem(P_j)$
    $v_{max} = \max \{v(P_j) : j \not\in J^*\}$
    $RCL = \{j \not\in J^* : v(P_j) \geq \alpha v_{max}\}$
end
return RCL

Figure 2. The construction procedure MakeRCL() for the set RCL
key = 0
while (key == 0) do
    RCL = MakeRCLSub(I'_j, I)
    if RCL ≠ ∅ then
        s = SelectDemand(RCL)
        I'_j = I'_j ∪ {s}
        I = I - {s}
    else
        key = 1
    end
end

Figure 3. The construction procedure for sub-problem P_j

In MakeRCL procedure, a set of demand points that can be assigned to candidate center j with k_j servers is determined.

\[
RCL = ∅ \\
a_{\text{max}} = \max \{a_i : i \in I, W(\lambda_j, k_j) \leq \tau, \lambda_j = \sum_{t \in I_j} f_t + f_i\}
\]

if (i \in I and \(a_i \geq a_{\text{max}}\) and \(W(\sum_{t \in I_j} f_t + f_i, k_j) \leq \tau\)) then
    RCL = RCL ∪ \{i\}
End

Figure 4. The MakeRCL procedure for sub-problem P_j

values of \(k_j (k_j = 1, \ldots, k_{\text{max}} \text{ and } \sum_j k_j \leq \bar{k})\), and calculate the covered population. The best solution for sub-problem P_j is then determined.

To solve the sub-problem P_j (with fixed \(k_j\)), we also use a modified version of GRASP. Fig. 3 shows the steps of the construction phase, and Fig. 4 shows the outline of the procedure Make RCL. In these procedures, \(I\) is defined as before and \(I'_j\) is the set of demand points assigned to center \(j\).

4. Numerical Example

We have solved three sets of problems with 20, 30, and 50 points, for different values of \(p\). Parameters are set to have the same values as in [39]. In addition to the local search within GRASP, we have considered implementing the program with two other local searches, LS1, and LS2. In LS1 for all selected service centers \(j\) with \(k_j\) servers, we try to move \(t = 1, \ldots, k_j\) of the servers to another center \(r\) with the aim of improving the population coverage. In LS2, for each selected center \(j\), we try to move \(t_1\) and \(t_2\) servers to two centers \(r_1\) and \(r_2\), respectively; again, in order to improve the objective function value. Table 2 shows the results for different values of \(\tau\) and \(p\). In this table, \(\bar{k}\) is set to be \(p\), and \(k_{\text{max}}\). The first column is the method used to solve the test problems, and the second column shows the values for \(\tau\). The entries in the table are the coverage percentage for different values of \(p\). These numbers are the averages of five runs.
The results in Table 2 indicate that as the number of servers, $p$, increases, so does the population percentage covered, as expected. In addition, for a fixed $p$, increasing $\tau$ improves the solutions. For a given $p$, however, there is no pronounced difference among different solution methods; although GRASP with the two added local searches, in most cases, show a slight improvement in coverage.

To evaluate the effect of $\bar{k}$ on the solution, we solved some test problems with different values of $\bar{k}$. Table 3 shows results for $\tau = 12.75$ and $k_{max} = 3$. Columns 3 through 12 show the coverage percentages for two values of $\bar{k} = 3, 4$. Coverage percentages are the averages of 5 runs. Note that if $\bar{k} < p$, then there is a center with no server which is not meaningful.

Table 3 depicts the results for fixed values of $k$. As can be seen, by increasing $\bar{k}$, the solutions are improved in some cases. The same is true for fixed values of $\bar{k}$ and different values of $p$. Furthermore, comparison of results in tables 2 and 3 shows that for small values of $p$ ($p = 1, 2$), increasing $\bar{k}$ ($\bar{k} = 3, 4$) usually improves the solution, but for $p = 3, 4$ the average coverage does not change. This is to be expected, since in Table 2 we have $k = p$; Thus, increasing $\bar{k}$ while keeping $p$ fixed would provide an opportunity to assign more servers to a center, and hence leads to the possibility of serving more population.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\tau$</th>
<th>$P = 1$</th>
<th>$P = 2$</th>
<th>$P = 3$</th>
<th>$P = 1$</th>
<th>$P = 2$</th>
<th>$P = 3$</th>
<th>$P = 4$</th>
<th>$P = 5$</th>
<th>$P = 1$</th>
<th>$P = 2$</th>
<th>$P = 3$</th>
<th>$P = 4$</th>
<th>$P = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRASP</td>
<td>10</td>
<td>71.45</td>
<td>90.1</td>
<td>99.62</td>
<td>66.14</td>
<td>83.8</td>
<td>97.11</td>
<td>98.53</td>
<td>100</td>
<td>45.31</td>
<td>63.16</td>
<td>70.95</td>
<td>77.94</td>
<td>84.13</td>
</tr>
<tr>
<td></td>
<td>12.75</td>
<td>77.05</td>
<td>90.06</td>
<td>100</td>
<td>70.74</td>
<td>83.91</td>
<td>96.19</td>
<td>98.53</td>
<td>100</td>
<td>47.5</td>
<td>65.14</td>
<td>74.19</td>
<td>79.81</td>
<td>86.17</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>82.61</td>
<td>90.02</td>
<td>100</td>
<td>74.41</td>
<td>83.91</td>
<td>96.63</td>
<td>98.53</td>
<td>100</td>
<td>48.81</td>
<td>66.87</td>
<td>73.54</td>
<td>81.37</td>
<td>86.60</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>85.68</td>
<td>90.02</td>
<td>100</td>
<td>77.87</td>
<td>91.26</td>
<td>98.53</td>
<td>99.78</td>
<td>100</td>
<td>49.1</td>
<td>66.47</td>
<td>74.28</td>
<td>81.27</td>
<td>86.59</td>
</tr>
<tr>
<td>GRASP + LS1</td>
<td>10</td>
<td>71.57</td>
<td>90.1</td>
<td>100</td>
<td>66.36</td>
<td>84.35</td>
<td>97.11</td>
<td>98.57</td>
<td>100</td>
<td>45.42</td>
<td>65.4</td>
<td>72.11</td>
<td>78.74</td>
<td>84.64</td>
</tr>
<tr>
<td></td>
<td>12.75</td>
<td>77.05</td>
<td>90.1</td>
<td>100</td>
<td>70.74</td>
<td>85</td>
<td>96.63</td>
<td>98.53</td>
<td>100</td>
<td>47.61</td>
<td>67.42</td>
<td>74.64</td>
<td>81.04</td>
<td>86.34</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>83.36</td>
<td>90.1</td>
<td>100</td>
<td>74.41</td>
<td>87.93</td>
<td>96.63</td>
<td>98.53</td>
<td>100</td>
<td>49.04</td>
<td>67.23</td>
<td>74.22</td>
<td>81.6</td>
<td>86.99</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>86.1</td>
<td>91.11</td>
<td>100</td>
<td>77.87</td>
<td>92.94</td>
<td>98.53</td>
<td>99.78</td>
<td>100</td>
<td>49.8</td>
<td>67.12</td>
<td>74.59</td>
<td>81.69</td>
<td>87</td>
</tr>
<tr>
<td>GRASP + LS2</td>
<td>10</td>
<td>71.45</td>
<td>98.1</td>
<td>100</td>
<td>66.14</td>
<td>95.61</td>
<td>98.53</td>
<td>98.53</td>
<td>100</td>
<td>45.31</td>
<td>63.16</td>
<td>72.15</td>
<td>77.94</td>
<td>84.13</td>
</tr>
<tr>
<td></td>
<td>12.75</td>
<td>77.05</td>
<td>99.62</td>
<td>100</td>
<td>70.74</td>
<td>97.25</td>
<td>98.53</td>
<td>98.53</td>
<td>100</td>
<td>47.5</td>
<td>65.14</td>
<td>74.19</td>
<td>79.81</td>
<td>86.17</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>82.61</td>
<td>100</td>
<td>100</td>
<td>74.41</td>
<td>97.25</td>
<td>98.53</td>
<td>99.12</td>
<td>100</td>
<td>48.81</td>
<td>66.87</td>
<td>73.54</td>
<td>81.37</td>
<td>86.60</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>85.68</td>
<td>100</td>
<td>100</td>
<td>77.87</td>
<td>91.26</td>
<td>98.53</td>
<td>99.78</td>
<td>100</td>
<td>49.1</td>
<td>66.47</td>
<td>74.28</td>
<td>81.27</td>
<td>86.59</td>
</tr>
<tr>
<td>GRASP + LS1 + LS2</td>
<td>10</td>
<td>71.57</td>
<td>98.1</td>
<td>100</td>
<td>66.36</td>
<td>86.69</td>
<td>98.53</td>
<td>98.82</td>
<td>100</td>
<td>45.42</td>
<td>65.4</td>
<td>72.11</td>
<td>78.74</td>
<td>84.64</td>
</tr>
<tr>
<td></td>
<td>12.75</td>
<td>77.05</td>
<td>100</td>
<td>100</td>
<td>70.74</td>
<td>85</td>
<td>98.53</td>
<td>98.53</td>
<td>100</td>
<td>47.61</td>
<td>67.42</td>
<td>74.64</td>
<td>81.04</td>
<td>86.34</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>83.36</td>
<td>100</td>
<td>100</td>
<td>74.41</td>
<td>87.93</td>
<td>98.53</td>
<td>99.12</td>
<td>100</td>
<td>49.04</td>
<td>67.23</td>
<td>74.22</td>
<td>81.6</td>
<td>86.99</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>86.1</td>
<td>95.06</td>
<td>100</td>
<td>77.87</td>
<td>92.94</td>
<td>98.53</td>
<td>99.78</td>
<td>100</td>
<td>49.8</td>
<td>67.12</td>
<td>74.59</td>
<td>81.69</td>
<td>87</td>
</tr>
</tbody>
</table>
Table 3. Computational results for different values of $\bar{k}$

<table>
<thead>
<tr>
<th>Method</th>
<th>$\bar{k}$</th>
<th>$m = n = 20$</th>
<th>$m = n = 30$</th>
<th>$m = n = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$P = 1$</td>
<td>$P = 2$</td>
<td>$P = 1$</td>
</tr>
<tr>
<td>GRASP</td>
<td>3</td>
<td>90.1</td>
<td>100</td>
<td>83.36</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>90.1</td>
<td>100</td>
<td>83.91</td>
</tr>
<tr>
<td>GRASP + LS1</td>
<td>3</td>
<td>90.1</td>
<td>100</td>
<td>84.24</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>90.1</td>
<td>100</td>
<td>84.64</td>
</tr>
<tr>
<td>GRASP + LS2</td>
<td>3</td>
<td>90.1</td>
<td>100</td>
<td>83.36</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>90.1</td>
<td>100</td>
<td>83.91</td>
</tr>
<tr>
<td>GRASP + LS1 + LS2</td>
<td>3</td>
<td>90.1</td>
<td>100</td>
<td>84.26</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>90.1</td>
<td>100</td>
<td>84.64</td>
</tr>
</tbody>
</table>

5. Conclusion

We first discussed some optimization models for locating emergency facilities and reviewed several successful implementations of these models. We then considered a queueing covering location problem and proposed its extension with an M/G/k system. Since we do not have a closed form formula for the waiting time and the problem is NP-hard, hence we opted to use the heuristic solution procedure GRASP. We also used two local search procedures to improve the solution. Numerical results, however, did not show a substantial improvement in the solution as a result of added local search procedures.

Acknowledgements

We would like to express our gratitude to the anonymous referees for their helpful comments.

References


Taghizadeh Kakhki, H. and Moeen Moghadas, F. (2010), A semidefinite programming relaxation for the queueing covering location problem with an M/G/1 system, pp. 231-236, in Proceedings of the European Workshop on Mixed Integer Nonlinear Programming, CRIM.
