Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 32 (2005) 15–33
computers & operations research
www.elsevier.com/locate/dsw

# Towards unified formulations and extensions of two classical probabilistic location models

Roberto D. Galvão[a], Fernando Y. Chiyoshi[a], Reinaldo Morabito[b,*]

[a] *Programa de Engenharia de Produção, COPPE/Federal University of Rio de Janeiro, 21945-970 Rio de Janeiro, RJ, Brazil*
[b] *Departamento de Engenharia de Produção, Federal University of São Carlos, 13565-905 São Carlos, SP, Brazil*

**Abstract**

We give a unified view of Daskin's Maximum Expected Covering Location Problem (MEXCLP) and ReVelle and Hogan's Maximum Availability Location Problem (MALP), identifying similarities and dissimilarities between these models and showing how they relate to each other. These models arise in the location of servers in congested emergency systems. An existing extension of MEXCLP is reviewed; we then develop an extension of MALP and give the corresponding mathematical formulation. These two extensions are obtained when the simplifying assumptions of the original models are dropped and Larson's hypercube model is embedded into local search methods. In this paper these methods are further enhanced by the use of simulated annealing. Computational results are given for problems available in the literature.


*Keywords:* Probabilistic location models; Congested emergency systems; Hypercube model; Simulated annealing

## 1. Introduction

Probabilistic location problems deal with the stochastic nature of real-world systems. In these systems some parameters, such as for example travel times, the location of clients, demand and the availability of servers are treated as random variables. The objective is to determine robust server/facility locations that optimize a given utility function, for a range of values of the parameters under consideration. According to Owen and Daskin [1], the corresponding models capture the stochastic aspects of the problems through the explicit consideration of the probability distributions of the random variables; some authors incorporate these distributions into standard mathematical

* Corresponding author. Fax:+55-16-260-8240.
*E-mail address:* morabito@power.ufscar.br (R. Morabito).


doi:10.1016/S0305-0548(03)00200-4

programming formulations, while others use them within a queuing framework. In this paper, we are particularly interested in the queuing framework.

A detailed review of probabilistic location problems can be found in Owen and Daskin [1]. Swersey [2], Chiyoshi et al. [3] and Brotcorne et al. [4] also examine some probabilistic models. Our interest lies in models that treat the availability of servers as a random variable, of particular importance for the location of emergency services. When modeling emergency systems the use of simplifying assumptions may allow the definition of mathematical programming models, for example through the definition of chance constraints. Situations in which the simplifying assumptions are not applicable lead to models based on spatially distributed queues, in which Larson's Hypercube Model [5,6] is of paramount importance.

The hypercube model, proposed by Larson [5] and studied by several authors (see Burwell et al. [7]; Swersey [2]), is an important tool for planning service systems, especially urban systems in which servers travel to offer some type of service to clients (server-to-customer service). The model treats geographical and temporal complexities of the region under study, based on the theory of spatially distributed queues. Basically, the idea is to expand the description of the state space of a queuing system with multiple servers, in order to represent each server individually and incorporate more complex dispatch policies.

To solve the model it is necessary to find the solution of a system of linear equations that provides the equilibrium probabilities of all possible states of the system (the *state probabilities*). These probabilities allow the computation of performance measures that are important for the management of the system, such as servers' workloads, dispatch frequencies and travel times, among others. Although applicable to *scenario planning*, the hypercube model is a descriptive model that, applied as a "stand alone" model, does not allow the direct solution of location problems.

Important models that treat the availability of servers as a random variable are the Maximum Expected Covering Location Problem (MEXCLP) of Daskin [8] and the Maximum Availability Location Problem (MALP) of ReVelle and Hogan [9]. In both models simplifying assumptions lead to the definition of mathematical programming models: the assumption that servers operate independently is a common feature to both models. Daskin also assumes that each server has the same busy probability. ReVelle and Hogan define two variations for MALP: MALPI, where the authors assume, as Daskin, that each server has the same busy probability, and MALPII, where they allow busy fractions to be different in the various sections of a region under consideration.

Batta et al. [10] suggest that an approximate way to relax the server independence assumption of MEXCLP is to use the hypercube correction factor developed by Larson [6]. This correction factor, applied to the MEXCLP objective function, leads to an "adjusted" model, which the authors call AMEXCLP. Recently, Saydam and Aytug [11] developed a genetic algorithm that combines MEXCLP with the hypercube approximation algorithm developed by Jarvis [12], in order to solve MEXCLP with increased accuracy. The fitness (objective) function of the genetic algorithm incorporates the hypercube methodology directly into the expected coverage calculations. Regarding MALP, Marianov and ReVelle [13] used results from queuing theory (an M/G/$s$/$s$ queuing system with $s$ servers in the neighborhood of a node and up to $s$ calls being serviced at the same time) to relax in an approximate way MALP's assumption of independence of server busy fractions, obtaining a more realistic model (which the authors named Queuing MALP or Q-MALP).

Batta et al. [10] also relaxed MEXCLP's simplifying assumptions by embedding the hypercube model into a single node vertex substitution heuristic procedure, seeking to determine a set of server locations that maximizes expected coverage. Galvão et al. [14] used the same approach to relax the simplifying assumptions of the MALPI model, seeking to maximize the population covered with a predefined reliability $\alpha$. In both cases the extended models are able to deal with server cooperation and the definition of busy fractions for each individual server, which reflects more precisely the situation in real-world systems. The idea of both papers is to reproduce conditions that are closer to those expected in practical applications.

In this paper, we give a unified view of the MEXCLP and MALPI models, identifying similarities and dissimilarities between these models and showing how they relate to each other. We also develop an extension of MALPI with the hypercube model embedded into it, giving the corresponding mathematical formulation. Even though this formulation, because of its complexity, is of little practical use, it has a structure that allows the development of heuristic solution methods. Finally, we use simulated annealing (SA) in an attempt to enhance the local searches developed for the extended MEXCLP and MALPI models. The results produced by the SA algorithm are particularly important for the extended MALPI model: since the coverage provided by different solutions is at great variance, in this case the better solutions produced by the SA methodology are of considerable practical importance.

This paper is organized as follows. In Section 2 models MEXCLP and MALPI are reviewed in detail. The basic assumptions of each model are stated and mathematical formulations of both models are given. A common notation is used in order to enable the comparison of these models. A relationship between the MEXCLP and MALPI objective functions is established and illustrated through an example. The experiences of Batta et al. [10] and Galvão et al. [14], where the hypercube model is embedded into vertex substitution procedures in order to relax the simplifying assumptions of MEXCLP and MALPI, respectively, are reviewed in Section 3. In this same section a mathematical formulation of the extended MALPI model is given. Computational results related to the SA procedure used in this paper and their analyses are treated in Section 4, which is followed by the Conclusions (Section 5).

## 2. The MEXCLP and MALPI probabilistic models: formulations and relationships

The MEXCLP and MALPI models are both probabilistic extensions of the Maximum Covering Location Problem (MCLP), their deterministic equivalent. The location of emergency services has as a common objective the provision of *coverage* to demand areas. The notion of *coverage* implies the definition of a *service distance* (*time*), which is the *critical distance* (*time*) beyond which a demand area is considered *not covered*. A demand area is therefore considered *covered* if it is within a predefined critical distance (say $S$) from at least one of the existing servers (facilities).

The simplest deterministic model defined for problems with covering constraints, the *Set Covering Problem* (*SCP*), provides coverage for every demand area under consideration. The provision of total coverage, however, may prove to be economically infeasible, in the sense that the number of servers required may not be compatible with the resources available to the decision-maker. MCLP was defined by Church and ReVelle [15] within this context. In this case the objective is to locate a number of facilities (say $p$ facilities) that is compatible with the resources available, such that the maximal possible population of a given geographical region is covered within $S$.

A limitation of the deterministic models is that they assume that servers are available when requested, which is not always true in practical situations. In non-congested systems, with little demand, the assumption is reasonable, but in congested systems, in which frequent calls for service may for example keep ambulances busy 20–30% of the time, the assumption is totally unjustifiable. Congestion in emergency services, which may cause the unavailability of servers within the critical distance when a call is placed, lead to the development of probabilistic covering models.

### 2.1. *The maximum expected covering location problem (MEXCLP)*

Daskin [8] defined the Maximum Expected Covering Location Problem (MEXCLP). MEXCLP's objective is to maximize the expected coverage of all demand areas under consideration. Daskin assumes that servers operate independently and that all servers have the same busy probability (*workload*) $\rho$. In his model Daskin allows more than one server to be situated in any given location.

Let $J$ be the set of demand areas. The probability that a demand area $j \in J$ is covered by at least one server, given that $k$ servers cover this area within the critical distance $S$, is given by $P$ [at least one server available within $S$] $= (1 - P$ [no server available within $S$]$) = 1 - \rho^k$.

Let $H_{j,k}$ be a random variable equal to the demand of area $j$ covered by an available server, given that $k$ servers cover this area. If $\phi_j$ is the number of calls per day originating in demand area $j$, $H_{j,k} = \phi_j$ with probability $(1-\rho^k)$, $H_{j,k} = 0$ with probability $\rho^k$. The expected value of $H_{j,k}$ is given by $E(H_{j,k}) = \phi_j\ (1-\rho^k)\ \forall j,k$. The increase in expected coverage in demand area $j$ when the number of servers that cover it is increased from $(k-1)$ to $k$ is given by $\Delta E(H_{j,k}) = E(H_{j,k}) - E(H_{j,k-1}) = \phi_j \rho^{k-1}(1-\rho),\ k = 1, 2, \ldots, p$.

Let $I$ ($|I| = n$) be the set of locations where servers may be stationed; $a_{ij} = 1$ if demand area $j$ is covered by a server located at $i \in I$ within critical distance $S$ ($a_{ij} = 0$ otherwise); $p$ is the number of servers to be located. Define variable $x_{jk} \in \{0, 1\}$ such that $x_{jk} = 1$ if demand area $j$ has at least $k$ servers within $S$, $x_{jk} = 0$ otherwise; finally let $y_i = 0, 1, 2, \ldots, p$ represent the number of servers located at $i \in I$. Using the definitions above MEXCLP may be formulated as an integer-programming problem:

$$\text{Maximize} \quad Z = \sum_{j \in J} \sum_{k=1}^{p} (1-\rho)\rho^{k-1}\phi_j x_{jk} \tag{1}$$

$$\text{Subject to} \quad \sum_{k=1}^{p} x_{jk} \leqslant \sum_{i \in I} a_{ij} y_i, \quad j \in J, \tag{2}$$

$$\sum_{i \in I} y_i \leqslant p, \tag{3}$$

$$y_i = 0, 1, \ldots, p, \quad i \in I, \tag{4}$$

$$x_{jk} \in \{0, 1\}, \quad \forall j, k. \tag{5}$$

In the formulation above the objective function maximizes the expected coverage, considering that up to $p$ servers may cover any given demand area within $S$. Restrictions (2) count the number of servers that cover demand area $j$ within $S$, for all $j \in J$. Constraint (3) sets at $p$ the upper limit on the number of servers to be located and restrictions (4) and (5) define the nature of the decision variables. Notice that restrictions (4) allow up to $p$ servers to be situated in any given location $i \in I$. Finally, as the objective function is concave in $k$ for each $j \in J$ (see Daskin [8]), it is not necessary to include in this formulation precedence constraints of the type $x_{jk} \leqslant x_{j(k-1)}$.

### 2.2. The maximum availability location problem I (MALPI)

In the definition of MALPI, ReVelle and Hogan [9] used a formula developed by Daskin [16] to estimate $\rho$, the common busy fraction they use for all servers. Let $\bar{t}$ be the mean service time, measured in hours, for a call originating in any demand area $j \in J$. It is possible to calculate $\rho$ as $\rho = \bar{t} \sum_{j \in J} \phi_j / 24p$, i.e. the busy fraction of each server is calculated dividing the mean number of daily hours of service needed by the system by the number of daily hours available, assuming that $p$ servers will be located.

The restriction that at least one server must be available within $S$ for any given demand area $j \in J$ with probability greater than or equal to $\alpha$ may be written in the following way: $P$ [at least one server available within $S$] $\geqslant \alpha \equiv$ $(1 - P$ [no server available within $S$]$) \geqslant \alpha = 1 - \rho^{\sum_{i \in I} a_{ij} y_i} \geqslant \alpha$, where $\sum_{i \in I} a_{ij} y_i$ is the number of servers available within $S$ of demand area $j \in J$. Or, taking logarithms, $\sum_{i \in I} a_{ij} y_i \geqslant d$, where $d = \lceil \log(1-\alpha)/\log \rho \rceil$, $\lceil u \rceil$ denoting the smallest integer greater than or equal to $u$.

From the equivalent linear expression obtained from the probabilistic constraint, it is possible to notice that each demand area $j \in J$ requires at least $d$ servers available within critical distance $S$ for it to be covered with reliability $\alpha$. In order to be able to maximize the number of calls serviced with reliability $\alpha$, it is necessary therefore to maximize the number of calls with at least $d$ servers available within $S$.

If we use variable $x_{jk}$ utilized in the definition of MEXCLP, the expression $\sum_{k=1}^{n} x_{jk}$ represents the number of times demand area $j \in J$ is covered within $S$. In order to maximize the number of calls covered with reliability $\alpha$, $\sum_{j \in J} \phi_j x_{jd}$ must be maximized. The mathematical formulation of MALPI can be finally written in the following way:

$$\text{Maximize} \quad Z = \sum_{j \in J} \phi_j x_{jd} \tag{6}$$

$$\text{Subject to} \quad \sum_{k=1}^{d} x_{jk} \leqslant \sum_{i \in I} a_{ij} y_i, \quad j \in J, \tag{7}$$

$$x_{jk} \leqslant x_{j(k-1)}, \quad j \in J, \quad k = 2, \ldots, d, \tag{8}$$

$$\sum_{i \in I} y_i = p, \tag{9}$$

$$x_{jk} \in \{0, 1\}, \quad j \in J, \quad k = 1, \ldots, d, \tag{10}$$

$$y_i \in \{0, 1\}, \quad i \in I. \tag{11}$$

Restrictions (7) guarantee that a demand area $j \in J$ is covered with reliability $\alpha$ if at least $d$ servers are available within $S$ from $j$. Constraints (8) express that, for a given demand area to be covered by $k$ servers, it must be covered by at least $(k-1)$ servers, for $2 \leqslant k \leqslant d$. These constraints, omitted in Daskin's model, are necessary in MALPI. Restriction (9) establishes that $p$ servers must be located, while constraints (10) and (11) define the binary nature of the decision variables. Notice that in the formulation of MALPI, as defined by ReVelle and Hogan [9], only one server may be situated in any location $i \in I$. The model remains valid, however, if more than one server is allowed in any location $i \in I$, as in MEXCLP; in this case constraints (11) would have to be replaced by constraints (4).

The formulation above is similar to that of MEXCLP, except of course for the objective function. Notice also the difference between constraints (2) and (7): while in MALPI $d$ servers within $S$ are needed to provide reliability $\alpha$ for demand area $j$, up to $p$ servers may provide coverage to any demand area $j \in J$ and contribute to the expected coverage expressed in the objective function of MEXCLP.

### 2.3. A relationship between the MEXCLP and MALPI objective functions

Since both Daskin [8] and ReVelle and Hogan [9] refer to the same system, specified in the same way, we may say that the two corresponding models, MEXCLP and MALPI, are just different ways of looking at the same phenomenon, and that there must be some sort of relationship between them. To the best of our knowledge this point has not been raised in the corresponding literature.

In order to see the nature of this relationship, we start with the coverage function of Daskin, observing that if a node $j$ is covered by $k$ servers, the contribution of this node to the expected coverage is, as shown in Section 2.1, $\phi_j\ (1-\rho^k)$. Denoting by $P_k$ the total population of the nodes covered by $k$ servers, the expected coverage (EC) of Daskin's model can be written as $\mathrm{EC} = \sum_{k=1}^{p} (1-\rho^k) P_k$.

In terms of the terminology of ReVelle and Hogan's model, we may say that each term of this expression consists of the population $P_k$ multiplied by the availability $(1-\rho^k)$ associated with the coverage the corresponding population is given. Instead of combining availability and population into a single figure, ReVelle and Hogan prefer to keep them apart, so that in their approach there is an underlying function relating availability to coverage; the desired level of availability must be specified beforehand, so that coverage can be determined. This function is a decreasing step function, with the steps being located at each availability level $(1-\rho^k)$. We may say that this function is downward cumulative, in the sense that a coverage that is $\alpha$-reliable is also $\alpha'$-reliable, for $\alpha' < \alpha$. With this fact in mind, we may write the $(1-\rho^k)$-reliable coverage (RC) as RC $(1-\rho^k) = \sum_{v=k}^{p} P_v$.

We illustrate the relationship between the two models using the 55-vertex network of Swain [17], with three servers located at nodes 2, 7 and 19, system workload of $\rho = 0.80$ and critical distance of $S = 15$, assuming that the basic hypotheses of both models are in force: servers operate independently, with the same workload. The relevant figures are shown in Table 1 below.

From this table it becomes clear where the main focus of each model lies: the total of the column of expected coverage (EC) for MEXCLP and a point of the curve RC $(1-\rho^k)$ for MALPI. The expected coverage of 2435 includes calls from all availability levels that are serviced within the critical time (distance), on the average. On the other hand, when we say that 3900 calls are given 0.49-reliable coverage, we are shifting our focus away from the calls that are kept waiting in excess

Table 1
Expected coverage and availability data: 55-vertex/3-server system

| $k$ | $(1-\rho^k)$ | $P_k$ | EC | RC |
|---|---|---|---|---|
| 1 | 0.20 | 460 | 92 | 5560 |
| 2 | 0.36 | 1200 | 432 | 5100 |
| 3 | 0.49 | 3900 | 1911 | 3900 |
| Expected coverage→ | | | 2435 | — |

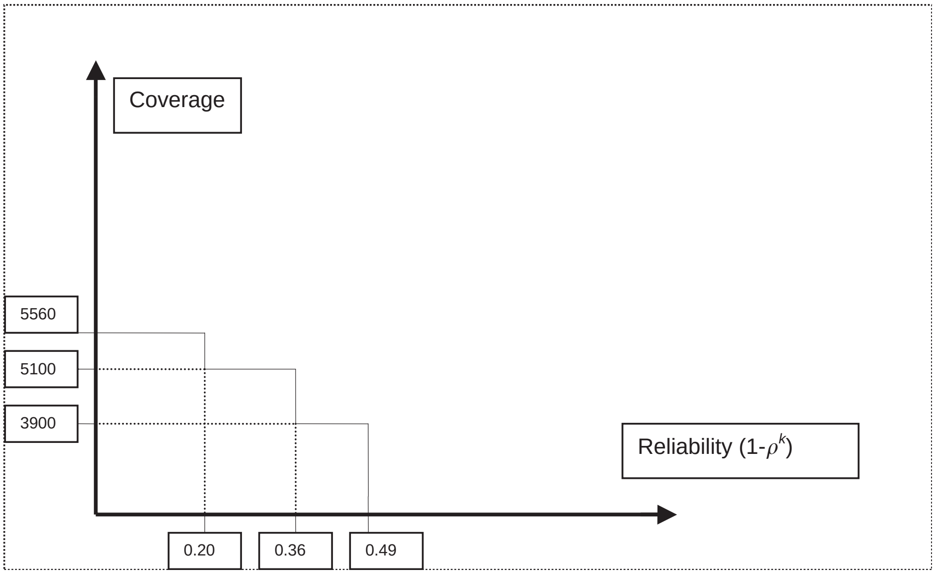


Fig. 1. Graphical representation of the coverage function: 55-vertex/3-server system.

of the critical time (in this case 51% of the calls), and totally overlooking the coverage that is provided at other availability levels. A graphical representation of the coverage function for this illustration is given in Fig. 1.

Note in Fig. 1 the meaning of the downward cumulative nature of the coverage function. For example, the 20% reliable coverage (given by one server) incorporates both the 36% coverage (given by two servers) and the 49% coverage (given by three servers).

The MALPI approach becomes meaningful when we are dealing with systems operating at high availability levels, as reliable emergency services generally do. If we have an availability of, say, 0.95, it seems reasonable to associate the quality of service to the coverage provided to the whole population, instead of computing the associated expected coverage. One should keep in mind that, in the long run, one out 20 calls will not be serviced within the critical time.

The availability-based concept of coverage has the additional advantage of being a non-technical, easy to grasp concept. That is perhaps the reason why an important system, the London Ambulance Service, sets its performance target based on this model: for life endangered calls, 50% of the calls must be serviced within 8 min, and 95% within 14 min (see http://www.londonambulance.nhs.uk/aboutus/facts/facts.html). In this respect Geandreau et al. [18] refer to 1973 American legislation that defines that 95% of the calls must be serviced within 10 min and cite that the city of Montreal requires that 70% of the calls be serviced within 7 min.

## 3. Extensions of MEXCLP and MALPI

The formulations of MEXCLP and MALPI given in Section 2 assume that servers operate independently and have the same workload $\rho$. When considered within the context of the hypercube model, each of the two hypotheses could be acceptable as an approximation. For example, if the system operates with high overall workload, the cooperation among servers would contribute to more homogeneous workloads; in this case the use of a common workload for all servers should not introduce grave distortions. On the other hand, if the overall system workload were low, the cooperation among servers would not be so important, making more acceptable the independence hypothesis. It is not difficult to see, however, that when considered jointly, the two hypotheses work in such a way that one weakens the other: when there is little cooperation, the relative independence of the servers makes non-homogeneous workloads an important issue; when there is cooperation, the independence hypothesis becomes questionable.

As already discussed, the absence of simplifying assumptions (which makes the modeling more adherent to real-world situations), often leads to the treatment of probabilistic location problems through spatially distributed queues. This is the approach we will follow in the remainder of this paper.

### 3.1. The relaxation of Daskin's hypotheses

Batta et al. [10] relax the following three hypotheses of Daskin's MEXCLP: (i) servers operate independently; (ii) all servers have the same workload $\rho$; (iii) the workloads do not vary with the location of the servers. The authors assume that cooperation among servers always happens in practice, given the need for a rapid response to emergency calls. This cooperation invalidates the independence hypothesis made by Daskin. Relaxing the three basic hypotheses of MEXCLP, assuming that the queuing system is in equilibrium and that the basic hypotheses of the hypercube model (see Larson [5]) hold in this case, Batta, Dolan and Krishnamurthy embed this model into a vertex substitution heuristic, seeking to determine the locations of the servers that will maximize expected coverage.

To fulfill their goal they use the hypercube model to compute expected coverage in each iteration of the algorithm; the iterative process is repeated until no further improvement in expected coverage is possible by making single node substitutions from the current set of server locations. This approach allows the use of workloads that are different for each server, reflecting their locations, and different dispatch policies, among other factors. It consequently increases the precision of the calculations.

Batta et al. compare, for the 55-vertex network used by Daskin, the following approaches: (i) heuristic and post-heuristic analyses of Daskin, followed by the post-IP analysis proposed by them, applied to the objective functions defined for MEXCLP and AMEXCLP; (ii) vertex substitution, with the hypercube model embedded into the heuristic developed by the authors (which we call EMEXCLP).

Chiyoshi et al. [19] observe that the three models compared by Batta, Dolan and Krishnamurthy are in fact not strictly comparable: by analyzing their objective functions it can be seen that, while both MEXCLP and AMEXCLP are restricted to coverage arising from unqueued calls, the embedding of the hypercube model into a vertex substitution heuristic takes into account unqueued as well as queued calls to predict expected coverage. Unless the system is operating at a very low overall workload or under very restrictive cover constraints, in which case no significant contribution to coverage is expected from queued calls, the models should produce different expected coverage due to the very nature of their objective functions.

Chiyoshi, Galvão and Morabito conclude that AMEXCLP, by relaxing the assumption that the servers operate independently, improves on the results produced by MEXCLP. On the other hand EMEXCLP, in addition to allowing the relaxation of the other two simplifying assumptions of MEXCLP, provides a way of dealing with queued calls, which AMEXCLP does not. This procedure is therefore more than just an improvement over the MEXCLP/AMEXCLP models: it has the capability of providing a more accurate description of the system.

### 3.2. An extension of the MALPI model

As in the case of MEXCLP, the hypotheses of MALPI can be relaxed with the use of the hypercube model. To achieve this relaxation it is initially necessary to redefine the location variable $y_i$ since, in order to introduce individual workloads for each server, it is no longer sufficient to know that there is a server located in $i \in I$: it is necessary to identify which server is located there. Define then location variable $y_{ik}$ such that $y_{ik} = 1$ if server $k$ is located at $i$, $y_{ik} = 0$ otherwise.

Using different workloads for each server and introducing Larson's correction factor to account for the non-independence among servers, the covering constraints take in this case the following form:

$$\left\{ 1 - \prod_{k=1}^{p} \rho_k^{\sum_{i \in I} a_{ij} y_{ik}} Q\left( p, \rho, \sum_{i \in I} \sum_{k=1}^{p} a_{ij} y_{ik} - 1 \right) \right\} \geqslant \alpha,$$

where $\rho_k$ is the workload of server $k$ and $Q$, Larson's correction factor, is a function of the number servers, the overall system workload and the coverage provided by the $p$ servers to a given demand area. When a given demand area $j$ is not covered by at least one server, $Q$'s third parameter becomes $-1$. As $Q$ is not defined for this value we make the convention that $Q(p, \rho, -1) = 1$.

Notice that, differently from the original formulation of ReVelle and Hogan [9], the use of a more detailed location variable $y_{ik}$ allows more than one server to be located at any $i \in I$. In order to remove this flexibility it would be necessary to add to the model restrictions of the type $\sum_{k=1}^{p} y_{ik} \leqslant 1,\ i \in I$.

When the hypotheses of MALPI are relaxed, we obtain what we call the *Extended Maximum Availability Location Problem* (EMALP). In this case the cover constraints incorporate parameters that depend, with a high degree of complexity, on the values of the problem's decision variables. Consequently, it would be difficult to use the cover constraints of this more general formulation in a mathematical programming model of the problem. These constraints allow, however, the computation of the coverage with reliability $\alpha$ corresponding to a given configuration of the location of the servers (see the appendix for an example of these computations in worksheet format).

### 3.3. A mathematical formulation for EMALP

The mathematical formulation of EMALP can be given by defining the decision variable $z_j$ such that $z_j = 1$ if demand area $j$ has at least one server available within $S$ with reliability $\alpha$, $z_j = 0$ otherwise:

$$\text{Maximize} \qquad Z = \sum_{j \in J} \phi_j z_j \tag{12}$$

Subject to

$$\left[ \left\{ 1 - \prod_{k=1}^{p} \rho_k^{\sum_{i \in I} a_{ij} y_{ik}} Q \left( p, \rho, \sum_{i \in I} \sum_{k=1}^{p} a_{ij} y_{ik} - 1 \right) \right\} - \alpha \right] z_j \geqslant 0, \quad j \in J, \tag{13}$$

$$\sum_{i \in I} \sum_{k=1}^{p} y_{ik} = p, \tag{14}$$

$$z_j \in \{0, 1\}, \quad j \in J, \tag{15}$$

$$y_{ik} \in \{0, 1\}, \;\; i \in I, \;\; k = 1, \ldots, p. \tag{16}$$

The model given by (12)–(16) is of little practical use since, as already observed, the covering constraints (13) incorporate parameters $(\rho_k, Q)$ that depend, with a high degree of complexity, on the values of the decision variables. The structure of the model allows, nevertheless, the development of heuristic methods for its solution. Galvão et al. [14] proposed a heuristic procedure for EMALP that is similar to the single node vertex substitution heuristic of Batta et al. [10].

The heuristic developed for EMALP was tested on problems available in the literature—the 55-vertex network of Swain [17] and the 100- and 150-vertex networks of Galvão and ReVelle [20]—for a range of values of $\alpha$, $\rho$, $p$ and $S$. The extensive computational results reported by Galvão et al. [14] in general confirm the expected behavior of the percentage of the population covered with reliability $\alpha$ as a function of $\rho$, $p$ and $S$. The computing times are very fast when the hypercube model is solved using Larson's [6] approximate method.

## 4. Computational results

We chose to use simulated annealing as an additional methodology to obtain results for the two extended models for the following reasons:

(i) We naturally expected that SA would do better than local search methods developed for these two models (at the expense of additional computational effort).

Table 2
Specification of the test problems

| Number of vertices/servers | EMEXCLP | | EMALP | | |
|---|---|---|---|---|---|
| | $\rho$ | $S$ | $\rho$ | $S$ | $\alpha$ |
| 55/5,10,15 | 0.50 | 15 | 0.50 | 15 | 0.90 |
| 100/5,10,15 | 0.50 | 60 | 0.50 | 80 | 0.90 |
| 150/5,10,15 | 0.50 | 80 | 0.50 | 90 | 0.90 |

(ii) Simulated annealing is a meta-heuristic that is usually easily implemented and generally less time-consuming than more sophisticated procedures such as tabu searches and genetic algorithms.
(iii) A previous experience of the authors with SA for solving the $p$-median problem produced good quality results in reasonable computing times.

Chiyoshi and Galvão [21] used the vertex substitution algorithm of Teitz and Bart [22] as the local search heuristic in their statistical analysis of simulated annealing (SA) applied to the $p$-median problem. As the underlying local search heuristic used when the MEXCLP and MALPI were extended is the same as in Chiyoshi and Galvão [21], it is only natural, given the authors' successful experimentation in that paper, that the same basic principles used by Chiyoshi and Galvão are used in the SA algorithm developed in this paper (with adaptations where necessary). Readers are therefore referred to the Chiyoshi and Galvão paper for a detailed description of the SA methodology used here.

Simulated Annealing was tested against the pure vertex substitution (VS) local search heuristic, using the three networks mentioned in Section 3.2 (the 55-, 100- and 150-vertex networks). For each network, problems with 5, 10 and 15 servers were tested. Two objective functions, namely the objective function based on expected coverage defined by Batta et al. [10], and the one based on the EMALP formulation given in Section 3.3, were evaluated. Each test problem, for each of the two objective functions, was solved for 20 different randomly chosen initial solutions, both through the VS and SA methodologies. The codes were written for a Pascal/Delphi compiler (V13.0). The tests were conducted on a 1.4 GHz. Pentium IV microcomputer, with 256 Mb of RAM. The specification of the test problems is given in Table 2.

The mid-sized problem of 100 vertices and 10 servers was used for test runs, in order to determine parameters such as the number of cycles and initial temperatures for the SA algorithm. It was found that the best solutions in 17 (out of 20) runs were obtained before the 10th cycle for the EMEXCLP model and in 19 runs before the 30th cycle for the EMALP model. The run lengths of 10 and 30 cycles for the EMEXCLP and EMALP models, respectively, were considered appropriate to produce evidence of the ability of the SA algorithm to find improved solutions for the test problems.

The initial temperatures for the two models were also determined after some experimentation with 100-vertex/10-server problem. An initial temperature of 20 was found to be appropriate to evaluate the EMEXCLP objective function. In the case of the EMALP model, due to its more irregular objective function, it was necessary to raise the initial temperature to 100.

Table 3
Detailed results for the EMEXCLP model

| Number of vertices | Number of servers | Coverage | | Computing time (s) | | SA to VS ratios | |
|---|---|---|---|---|---|---|---|
| | | VS | SA | VS | SA | Coverage | Comp. time |
| 55 | 5 | 5069 | 5069 | 23.81 | 83.37 | 1.0000 | 3.50 |
| | 10 | 5870 | 5870 | 278.22 | 814.45 | 1.0000 | 2.93 |
| | 15 | 6177 | 6177 | 1423.41 | 3919.22 | 1.0000 | 2.75 |
| 100 | 5 | 912 | 913 | 49.50 | 225.33 | 1.0011 | 4.55 |
| | 10 | 1376 | 1379 | 642.34 | 1894.83 | 1.0022 | 2.95 |
| | 15 | 1713 | 1714 | 3064.23 | 7238.17 | 1.0006 | 2.36 |
| 150 | 5 | 2868 | 2869 | 112.44 | 474.59 | 1.0003 | 4.22 |
| | 10 | 5870 | 5870 | 278.22 | 814.45 | 1.0000 | 2.93 |
| | 15 | 6099 | 6100 | 5868.70 | 14093.54 | 1.0002 | 2.40 |

Table 4
Detailed results for the EMALP model

| Number of vertices | Number of servers | Coverage | | Computing time (s) | | SA to VS ratios | |
|---|---|---|---|---|---|---|---|
| | | VS | SA | VS | SA | Coverage | Comp. time |
| 55 | 5 | 5150 | 5150 | 10.00 | 152.20 | 1.0000 | 15.22 |
| | 10 | 6070 | 6210 | 166.34 | 1776.11 | 1.0231 | 10.68 |
| | 15 | 6350 | 6370 | 739.61 | 8952.58 | 1.0031 | 12.10 |
| 100 | 5 | 1236 | 1370 | 28.76 | 386.16 | 1.1084 | 13.43 |
| | 10 | 1874 | 1965 | 392.09 | 3601.14 | 1.0486 | 9.18 |
| | 15 | 2145 | 2187 | 1667.33 | 13641.06 | 1.0196 | 8.18 |
| 150 | 5 | 4313 | 4313 | 59.02 | 858.73 | 1.0000 | 14.55 |
| | 10 | 5563 | 7315 | 745.62 | 7464.81 | 1.3149 | 10.01 |
| | 15 | 7711 | 8483 | 3255.16 | 29280.36 | 1.1001 | 8.99 |

The SA algorithm was finally set to run with temperature adjustments at every other cycle, with an initial temperature of 20 and a run length of 10 cycles for EMEXCLP and with an initial temperature of 100 and a run length of 30 cycles for EMALP.

Results for each of the 9 problems of Table 2 are shown in Table 3 (for the EMEXCLP model) and Table 4 (for the EMALP model). In each line of these tables we show, in the "Coverage" column, the best solution obtained after the 20 runs made using different randomly chosen initial solutions, for both the VS and SA methodologies. The computing times are the total times for the 20 runs in both cases. The two last columns of these tables show the SA to VS ratios corresponding to coverage and computing times.

Table 5
Solutions for the EMEXCLP model: 55-vertex/10-server problem

| Rank | Coverage | Freq. of sol. | | Rank × freq. | |
|---|---|---|---|---|---|
| | | VS | SA | VS | SA |
| 1 | 5870 | 5 | 16 | 5 | 16 |
| 2 | 5869 | 1 | 3 | 2 | 6 |
| 3 | 5868 | 4 | 1 | 12 | 3 |
| 4 | 5867 | 2 | | 8 | 0 |
| 5 | 5864 | 1 | | 5 | 0 |
| 6 | 5863 | 2 | | 12 | 0 |
| 7 | 5859 | 3 | | 21 | 0 |
| 8 | 5855 | 1 | | 8 | 0 |
| 9 | 5854 | 1 | | 9 | 0 |
| Average rank→ | | | | 4.10 | 1.25 |

When we observe the “SA to VS Ratios” columns of Table 3, there is no evidence of significant improvement in coverage brought by SA for the EMEXCLP model. The maximum improvement is less than 1%, and the computing times can be up to 4.55 times higher when simulated annealing is used instead of the local search heuristic. The situation is however quite different for the EMALP model (Table 4). SA found considerably better solutions than the VS heuristic (31% better in one specific case), suggesting that, depending on the computing effort one is willing to spend, SA may be an attractive alternative in this case.

Besides looking at comparative figures based on the best of 20 solutions, it may be worth comparing the results based on the whole set of solutions for each problem. To that effect two characteristics were taken into account: the ability to obtain the best solution and the overall performance of each of the methods. The results for each problem are organized as shown in Table 5, which refers to the 55-vertex/10-server problem solved using the EMEXCLP objective function. The best solution was obtained 5 times (out of possible 20) by the VS methodology and 16 times by the SA algorithm. The overall performance of each method is measured by the average rank of the solutions produced by each of the methods.

A very different picture emerges from the data obtained when the EMALP model was used. For the 55-vertex/5-server problem, for example, 18 different solutions were obtained, with coverage ranging from 290 to 5150 (see Table 6). In this model the objective function seems to be very irregular, with a large number of local optima significantly different from each other.

A summary of the results obtained for the set of 9 test problems using the EMEXCLP model is shown in Table 7. In this table the third column (# Sol.) represents the total number of different solutions obtained by the two methods, VS and SA (the maximum possible number of different solutions is 40, 20 for each method).

In terms of the ability to obtain the best solution, the SA method outperformed the VS method in all but one test problem (the 150-vertex/5-server problem). In addition, there were five problems in which the best solution was found by SA method alone. In terms of the overall performance, as measured by the average rank of the solutions, the SA method performed consistently better than the

Table 6
Solutions for the EMALP model: 55-vertex/5-server problem

| Rank | Coverage | Freq. of sol. | | Rank × freq. | |
|---|---|---|---|---|---|
| | | VS | SA | VS | SA |
| 1 | 5150 | 1 | 8 | 1 | 8 |
| 2 | 5010 | 1 | 2 | 2 | 4 |
| 3 | 4970 | 2 | 1 | 6 | 3 |
| 4 | 4910 | | 1 | 0 | 4 |
| 5 | 4860 | | 1 | 0 | 5 |
| 6 | 4850 | 3 | 3 | 18 | 18 |
| 7 | 4810 | 1 | 1 | 7 | 7 |
| 8 | 4700 | 1 | | 8 | 0 |
| 9 | 4550 | 2 | 2 | 18 | 18 |
| 10 | 4490 | 1 | | 10 | 0 |
| 11 | 4460 | 1 | | 11 | 0 |
| 12 | 3510 | 1 | | 12 | 0 |
| 13 | 3350 | 1 | | 13 | 0 |
| 14 | 3270 | | 1 | 0 | 14 |
| 15 | 2040 | 2 | | 30 | 0 |
| 16 | 1530 | 1 | | 16 | 0 |
| 17 | 990 | 1 | | 17 | 0 |
| 18 | 290 | 1 | | 18 | 0 |
| Average rank→ | | | | 9.35 | 4.05 |

Table 7
Summary of results, EMEXCLP model

| Nodes | Servers | # Sol | Freq. of best sol. | | Average rank | |
|---|---|---|---|---|---|---|
| | | | VS | SA | VS | SA |
| 55 | 5 | 3 | 5 | 15 | 2.50 | 1.45 |
| | 10 | 9 | 5 | 16 | 4.10 | 1.25 |
| | 15 | 10 | 5 | 9 | 4.45 | 2.05 |
| 100 | 5 | 4 | 0 | 4 | 3.75 | 2.65 |
| | 10 | 16 | 0 | 2 | 9.45 | 4.00 |
| | 15 | 22 | 0 | 2 | 12.40 | 5.90 |
| 150 | 5 | 3 | 6 | 6 | 2.20 | 2.15 |
| | 10 | 15 | 0 | 1 | 8.75 | 7.20 |
| | 15 | 15 | 0 | 2 | 8.15 | 5.40 |

VS method. Closeness between the different solutions was observed throughout for the EMXECLP model, suggesting that its objective function is topped by an easy to reach, plateau-like, slightly irregular surface, with many local optima very close to each other.

Table 8
Summary of results, EMALP model

| Nodes | Servers | # Sol | Freq. of best sol. | | Average rank | |
|---|---|---|---|---|---|---|
| | | | VS | SA | VS | SA |
| 55 | 5 | 18 | 1 | 8 | 9.35 | 4.05 |
| | 10 | 25 | 0 | 1 | 17.05 | 5.15 |
| | 15 | 18 | 0 | 2 | 11.20 | 3.30 |
| 100 | 5 | 24 | 0 | 3 | 15.40 | 4.35 |
| | 10 | 34 | 0 | 1 | 24.45 | 8.85 |
| | 15 | 34 | 0 | 2 | 24.25 | 6.75 |
| 150 | 5 | 24 | 3 | 9 | 14.15 | 3.25 |
| | 10 | 39 | 0 | 2 | 29.45 | 9.60 |
| | 15 | 40 | 0 | 1 | 29.95 | 11.05 |

Table 9
SA to VS computing time ratio statistics for the test problems

| ↓Statistics/model→ | EMEXCLP | EMALP |
|---|---|---|
| Average | 3.2 | 11.4 |
| Minimum | 2.4 | 8.2 |
| Maximum | 4.6 | 15.2 |

A different scenario emerged when the EMALP model was used. A summary of the results obtained for the set of 9 test problems using the EMALP model is shown in Table 8.

In the case of EMALP the SA method shows the ability to produce better solutions, which tend to be concentrated closer to the best solution. In fact, it can be said that the performance of the SA method is much better for the EMALP model (when we compare its performance with that obtained using the MEXCLP model), because in this case the VS method found the best solution in two problems only, both with five servers. Since the coverage provided by different solutions can be at great variance in the case of EMALP, the better solutions produced by the SA method are of considerable practical importance.

In relation to computing times, in addition to the data presented in Tables 3 and 4, we show the behavior of the SA algorithm over the complete set of the test problems. Aggregated data related to the SA to VS computing time ratio are shown in Table 9.

## 5. Conclusions

In this paper we give a unified view of two classical probabilistic location models: MEXCLP of Daskin [8] and MALPI of ReVelle and Hogan [9], identifying similarities and dissimilarities between them and showing how they relate to each other. Both MEXCLP and MALPI refer to the same system: by combining population and availability data into a single figure we obtain the coverage of the MEXCLP model; on the other hand, if we prefer to single out one point of the function coverage versus availability, the coverage of the MALPI model is obtained.

We studied extensions of these two basic models: the extension of MEXCLP developed by Batta et al. [10] and the extension of MALPI developed by us. These extensions are obtained when the underlying simplifying assumptions of the original models—independence among servers and a common workload for all servers—are dropped and Larson's hypercube model is embedded into single node vertex substitution local search heuristics.

The computational effort of this paper was aimed at finding evidence of the ability of SA to improve on the solutions obtained via the vertex substitution (VS) heuristics previously developed (Batta et al. [10]; Galvão et al. [14]) for the extended models EMEXCLP and EMALP, for a set of test problems corresponding to networks available in the literature.

In spite of the limited computational experience presented in Section 4, the SA algorithm outperformed the VS heuristic as expected, in terms of both the ability to obtain the best solution and the overall performance of the two methodologies, the latter measured by the average rank of the solutions produced by each of the two methods. Closeness between the different solutions, observed throughout for the EMEXCLP model, suggests that the differences observed between the two methodologies may not be of practical importance in this case. In the case of EMALP, however, since the coverage provided by different solutions are at great variance, the better solutions produced by the SA methodology are of considerable practical importance and justify the additional computing effort required by the SA algorithm.

Other meta-heuristics, such as for example tabu search and genetic algorithms, may evidently be used to find solutions to the extended models. Examples of the use of tabu search and genetic algorithms in similar contexts can be found in Gendreau et al. [18,23] and Saydam and Aytug [11]. The comparison of different meta-heuristics for extensions of MEXCLP and MALPI is however beyond the scope of the present paper. We leave this as a suggestion for future research.

## Acknowledgements

The authors would like to thank two anonymous referees for their useful comments and suggestions. This research was supported by the Brazilian National Research Council (CNPq grants 471919/01-2, 522973/95-7) and by the Research Foundation of the state of Rio de Janeiro (FAPERJ grant E26/151.934/2000).

## Appendix. Computation of coverage in worksheet format for a given configuration of servers

In this appendix we show the detailed calculations, in worksheet format, of the coverage obtained for a given configuration of servers. The calculations shown in Table 10 correspond to the 55-vertex network of Swain [17]. We use a critical distance of $S = 15$, a reliability $\alpha = 0.50$ and 3 servers located in atoms 2, 7 and 19. The system workload is 0.80 and the workloads of the units 1, 2 and 3, obtained from the approximate solution of the corresponding hypercube model, are respectively

Table 10
Detailed computation of coverage for a given configuration of servers, 55-vertex network of Swain [17]

| Atom | Pop | Distance from unit | | | Covering units | | | Total units | Prob not available | | | Factor $Q$ | Prob available | Coverage |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 | | 1 | 2 | 3 | | | |
| 1 | 710 | 3.16 | 8.25 | 13.45 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 710 |
| 2 | 620 | 0.00 | 5.10 | 10.63 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 620 |
| 3 | 560 | 4.47 | 4.24 | 6.40 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 560 |
| 4 | 390 | 3.00 | 6.40 | 13.04 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 390 |
| 5 | 350 | 4.24 | 8.94 | 14.87 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 350 |
| 6 | 210 | 7.62 | 8.25 | 15.52 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 210 |
| 7 | 200 | 5.10 | 0.00 | 7.28 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 200 |
| 8 | 190 | 3.16 | 6.33 | 9.43 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 190 |
| 9 | 170 | 5.00 | 7.81 | 14.77 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 170 |
| 10 | 170 | 11.00 | 13.00 | 20.25 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 170 |
| 11 | 160 | 5.66 | 10.30 | 16.28 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 160 |
| 12 | 150 | 24.19 | 21.19 | 13.93 | 0 | 0 | 1 | 1 | 1.0000 | 1.0000 | 0.7635 | 1.0000 | 0.2365 | 0 |
| 13 | 140 | 5.38 | 10.44 | 15.62 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 140 |
| 14 | 120 | 28.28 | 27.02 | 20.22 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 15 | 120 | 8.94 | 5.83 | 12.04 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 120 |
| 16 | 110 | 19.03 | 18.97 | 13.60 | 0 | 0 | 1 | 1 | 1.0000 | 1.0000 | 0.7635 | 1.0000 | 0.2365 | 0 |
| 17 | 100 | 18.03 | 14.87 | 7.62 | 0 | 1 | 1 | 2 | 1.0000 | 0.8066 | 0.7635 | 0.6180 | 0.6194 | 100 |
| 18 | 100 | 12.04 | 7.00 | 8.60 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 100 |
| 19 | 90 | 10.63 | 7.28 | 0.00 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 90 |
| 20 | 90 | 18.44 | 19.03 | 26.17 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 21 | 90 | 20.00 | 21.59 | 28.86 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 22 | 80 | 16.76 | 15.00 | 8.25 | 0 | 1 | 1 | 2 | 1.0000 | 0.8066 | 0.7635 | 0.6180 | 0.6194 | 80 |
| 23 | 80 | 15.62 | 11.40 | 5.38 | 0 | 1 | 1 | 2 | 1.0000 | 0.8066 | 0.7635 | 0.6180 | 0.6194 | 80 |
| 24 | 80 | 23.77 | 19.31 | 21.26 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 25 | 80 | 14.87 | 13.00 | 19.23 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 80 |
| 26 | 70 | 19.00 | 14.04 | 14.42 | 0 | 1 | 1 | 2 | 1.0000 | 0.8066 | 0.7635 | 0.6180 | 0.6194 | 70 |
| 27 | 60 | 24.52 | 25.08 | 20.00 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 28 | 60 | 22.67 | 18.44 | 12.21 | 0 | 0 | 1 | 1 | 1.0000 | 1.0000 | 0.7635 | 1.0000 | 0.2365 | 0 |
| 29 | 60 | 11.66 | 7.07 | 3.61 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 60 |
| 30 | 60 | 9.22 | 8.54 | 5.10 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 60 |
| 31 | 60 | 8.54 | 3.61 | 5.10 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 60 |
| 32 | 50 | 13.34 | 14.42 | 11.18 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 50 |
| 33 | 50 | 13.15 | 12.37 | 7.07 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 50 |
| 34 | 50 | 6.71 | 9.43 | 10.20 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 50 |
| 35 | 50 | 23.26 | 19.42 | 22.80 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 36 | 50 | 15.65 | 12.04 | 16.55 | 0 | 1 | 0 | 1 | 1.0000 | 0.8066 | 1.0000 | 1.0000 | 0.1934 | 0 |
| 37 | 50 | 17.09 | 20.25 | 27.30 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 38 | 40 | 16.55 | 18.44 | 15.65 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 39 | 40 | 25.50 | 28.43 | 26.40 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 40 | 40 | 22.47 | 26.93 | 28.00 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 41 | 40 | 11.66 | 11.04 | 18.03 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 40 |
| 42 | 40 | 2.83 | 4.24 | 11.18 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 40 |
| 43 | 40 | 11.40 | 15.23 | 16.03 | 1 | 0 | 0 | 1 | 0.8299 | 1.0000 | 1.0000 | 1.0000 | 0.1701 | 0 |
| 44 | 40 | 7.00 | 12.04 | 16.13 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 40 |
| 45 | 30 | 9.49 | 11.31 | 10.05 | 1 | 1 | 1 | 3 | 0.8299 | 0.8066 | 0.7635 | 0.4214 | 0.7847 | 30 |

Table 10 (*continued*)

| Atom | Pop | Distance from unit | | | Covering units | | | Total units | Prob not available | | | Factor $Q$ | Prob available | Coverage |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 | | 1 | 2 | 3 | | | |
| 46 | 30 | 13.60 | 18.25 | 20.40 | 1 | 0 | 0 | 1 | 0.8299 | 1.0000 | 1.0000 | 1.0000 | 0.1701 | 0 |
| 47 | 30 | 9.22 | 13.89 | 19.80 | 1 | 1 | 0 | 2 | 0.8299 | 0.8066 | 1.0000 | 0.6180 | 0.5863 | 30 |
| 48 | 30 | 19.11 | 16.64 | 22.14 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 49 | 30 | 20.59 | 19.65 | 26.17 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 50 | 30 | 20.62 | 25.24 | 31.14 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 51 | 30 | 27.07 | 28.16 | 35.35 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 52 | 20 | 24.35 | 29.41 | 34.00 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 53 | 20 | 14.87 | 19.42 | 25.46 | 1 | 0 | 0 | 1 | 0.8299 | 1.0000 | 1.0000 | 1.0000 | 0.1701 | 0 |
| 54 | 20 | 22.82 | 24.84 | 21.63 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| 55 | 20 | 16.40 | 20.13 | 20.10 | 0 | 0 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.0000 | 0 |
| | 6400 | | | | | | | | | | | | | 5100 |

0.8299, 0.8066 and 0.7635. Larson's correction factor $Q$ takes the values of 1, 0.6180 and 0.4214 for 0 or 1, 2 and 3 servers available within $S$, respectively.

The columns of Table 10 are self-explanatory. The initial 5 columns show each atom with its respective population, and the distance of the atom from each of the 3 available servers. Comparing these distances with the value of 15 it is possible to determine which units cover the atom and the total number of covering units (next four columns). The probability of each unit not being available is given by the respective workloads (next 3 columns); Larson's corresponding $Q$ factor is shown in the following column. The complement of the multiplication of the last four columns described above gives the probability of at least one server being available within $S$ from the atom. If this probability exceeds the reliability $\alpha$ defined for the problem ($\alpha = 0.50$ in our example) the atom is covered; otherwise it is not. The covered population is shown in the last column. In our example 79.7% of the population is covered within a distance of 15 with reliability of 50%.

## References


[1] Owen SH, Daskin MS. Strategic facility location: a review. European Journal of Operational Research 1998;111: 423–47.

[2] Swersey AJ. The deployment of police, fire and emergency medical units. In: Pollock SM, et al., editors. Handbooks in OR & MS 6. Amsterdam: Elsevier Science B.V., 1994. p. 151–200.

[3] Chiyoshi FY, Galvão RD, Morabito R. O uso do modelo hipercubo na solução de problemas de localização probabilísticos (The use of the hypercube model in the solution of probabilistic location problems). Gestão & Produção 2000;7:146–74.

[4] Brotcorne L, Laporte G, Semet F. Ambulance location and relocation models. European Journal of Operational Research 2003;147:451–63.

[5] Larson RC. A hypercube queuing model for facility location and redistricting in urban emergency services. Computers and Operations Research 1974;1:67–95.

[6] Larson RC. Approximating the performance of urban emergency service systems. Operations Research 1975;23: 845–68.

[7] Burwell TH, Jarvis JP, McKnew MA. Modeling co-located servers and dispatch ties in the hypercube model. Computers and Operations Research 1993;20:113–9.
[8] Daskin MS. A maximum expected covering location model: formulation, properties and heuristic solution. Transportation Science 1983;17:48–70.
[9] ReVelle CS, Hogan K. The maximum availability location problem. Transportation Science 1989;23:192–200.
[10] Batta R, Dolan JM, Krishnamurthy NN. The maximal expected covering location problem: revisited. Transportation Science 1989;23:277–87.
[11] Saydam C, Aytug H. Accurate estimation of expected coverage: revisited. Socio-Economic Planning Sciences 2003;37:69–80.
[12] Jarvis JP. Approximating the equilibrium behavior of multi-server loss systems. Management Science 1985;31: 235–9.
[13] Marianov V, ReVelle C. The queuing maximal availability location problem: a model for the siting of emergency vehicles. European Journal of Operational Research 1996;93:110–20.
[14] Galvão RD, Chiyoshi FY, Espejo LGA, Rivas MPA. Solução do problema de localização de máxima disponibilidade utilizando o modelo hipercubo (Solution of the maximum availability location problem using the hypercube model). Pesquisa Operacional 2003;23:61–78.
[15] Church R, ReVelle CS. The maximal covering location problem. Papers of the Regional Science Association 1974;32:101–18.
[16] Daskin MS. Application of an expected covering model to emergency medical service system design. Decision Sciences 1982;13:416–39.
[17] Swain R. A decomposition algorithm for a class of facility location problems. PhD thesis, Cornell University, Ithaca, NY, 1971.
[18] Gendreau M, Laporte G, Semet F. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. Parallel Computing 2001;27:1641–53.
[19] Chiyoshi FY, Galvão RD, Morabito R. A note on solutions to the maximal expected covering location problem. Computers and Operations Research 2003;30:87–96.
[20] Galvão RD, ReVelle CS. A Lagrangean heuristic for the maximal covering location problem. European Journal of Operational Research 1996;88:114–23.
[21] Chiyoshi FY, Galvão RD. A statistical analysis of simulated annealing applied to the $p$-median problem. Annals of Operations Research 2000;96:61–74.
[22] Teitz MB, Bart P. Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research 1968;16:955–61.
[23] Gendreau M, Laporte G, Semet F. Solving an ambulance location model by tabu search. Location Science 1997;5: 75–88.

**Roberto Diéguez Galvão** is a Full Professor in the Post-Graduate Department of Production Engineering at the Federal University of Rio de Janeiro. His research interests include optimization of discrete location and distribution models and the embedding of these models into Spatial Decision Support Systems. He has several publications in these areas in the last two decades.

**Fernando Yassuo Chiyoshi** is an Associate Professor in the Post-Graduate Department of Production Engineering at the Federal University of Rio de Janeiro. His research interests include queueing theory, simulation in meta heuristics, particularly simulated annealing. He has some publications in these areas in the recent years.

**Reinaldo Morabito** is an Associate Professor in the Post-Graduate Department of Production Engineering at the Federal University of São Carlos, Brazil. His research interests are in the fields of operations management and operations research. He has published articles in cutting and packing problems, queuing networks applied to manufacturing systems, combinatorial optimization, logistics and transportation planning.