Probabilistic Maximal Covering Location-Allocation Models with Constrained Waiting Time or Queue Length for Congested Systems

by

Vladimir Marianov
Department of Electrical Engineering
The Catholic University of Chile, Santiago, CHILE

and

Daniel Serra[1]
Department of Economics, IET and GRES
Universitat Pompeu Fabra, Barcelona, SPAIN

Abstract

When dealing with the design of service networks, such as health and EMS services, banking or distributed ticket selling services, the location of service centers has a strong influence on the congestion at each of them, and consequently, on the quality of service. In this paper, several models are presented to consider service congestion. The first model addresses the issue of the location of the least number of single-server centers so that all the population is served within a standard distance, and nobody stands on line for a time longer than a given time-limit, or with more than a predetermined number of other clients. We then formulate several maximal coverage models, with one or more servers per service center. A new heuristic is developed to solve the models and tested in a 30-nodes network.

[1] This research has been possible thanks to grants by NORTEL - External Research, FONDECYT (the chilean state scientific research fund) and the Fundación Banco Bilbao Vizcaya, Spain.

Introduction

When dealing with the design of service networks, as health, banking or distributed ticket selling services, the location of service centers has a strong influence on the congestion at each of them, and consequently, on the quality of service. Examples of these types of systems are the primary health care center services, which have centers with one physician attending general cases and referring those cases he/she can not cure, to a specialist. These centers must be located in such a way that they could be reached from any demand point within a reasonably short time, and, once a patient has arrived to the center, his/her waiting time should be as short as possible, since waiting time is an important determinant of the perception of service quality. Other examples of such services are banks, with offices distributed over a geographical area, where clients are standing on lines, waiting to be served. Or distribution centers, where trucks arrive to deliver their load. The presence of more trucks than servers at a center, can make it easily congested.

Several models have been presented which are explicitly intended for the design of spatial queueing systems. In general, though, these models are oriented to emergency systems, in which servers travel to the site of the emergency, as opposed to systems in which servers are fixed, as health care centers. Some of them, assume a single server in the region under study. Berman, Larson and Chiu (1985), for example, develop a heuristic algorithm to locate optimally one server on a congested network. They formulate what they call the Stochastic Queue Median. A model by Batta (1988), considers the situation in which there might be a selective rejection of calls by the dispatcher. The model is presented together with a greedy heuristic procedure for location of the server. Batta, Larson and Odoni (1988) present a model and an algorithm for locating one server when there are calls of different priorities. Batta (1989) presents a model to study the effect of using expected service time dependent queueing disciplines on optimal location of a single server.

Models assuming single servers are, in some cases, used as building blocks for algorithms that locate more than one server. Based on the one-server location algorithm of Berman, Larson and Chiu (1985), Berman, Larson and Parkan (1987) develop two heuristics for locating *p* servers on a congested network. At each step, both use a procedure called Mean Time Calibration (Larson and Odoni, 1981), which in turn includes solving the exact hypercube. This makes this algorithm suitable for systems with only a few servers. After the Mean Time Calibration is performed, the first heuristic uses the 1-median to improve the location of each one of the servers, while the second one uses the Stochastic Queue Median for the same purposes. All of these models are nonlinear.

Some of the models have considered also the problem of districting, or determination of optimal service territories. Berman and Larson (1985) solve the problem of districting for a two-server network in the presence of queueing. Given the locations of two servers on a congested network, a nonlinear model and a heuristic algorithm are developed to determine the optimal service territories of each server. Each district behaves as a M/G/1 system, with FIFO queues. Optimality means, in this case, minimum average response time to a random customer. No interaction exists between districts.

Later, Berman and Mandowsky (1986), use the Stochastic Queue Median, combined with this 2-server districting algorithm, to develop a general location - districting iterative algorithm for two units, and for *n*-nodes, *m*-server networks. In the first case, the best districting is found for a first location choice, followed by a re - location of both servers given fixed districts. This procedure is iterated as needed. In the case of *m* servers, at each step, two servers are optimally located and their optimal service areas found, while all the remaining units stay at a fixed location. No interaction exists between districts.

In general, a fairly large computational effort is required if these models are to be used for location of servers. All of the models are nonlinear, heuristic, and their

objective is to minimize expected response time of servers that travel to the site of an emergency. Also, all models use approximations in order to model the system.

On the other hand, optimization models for location, derived from the Location Set Covering Problem (LSCP, Toregas *et al*, 1974), the *p* - median (Hakimi, 1964, ReVelle and Swain, 1970), and the Maximal Covering Location Problem (MCLP, Church and ReVelle, 1974), are usually linear, and can be solved to optimality in a reasonable time. To deal with congestion, optimization models typically include a capacity constraint, which forces the demand for service at each center to be smaller than its maximum capacity. In these constraints, the maximum capacity is usually given by the number of servers at the center, or by some estimation of the maximum number of users that the center can serve at the same time. On the other hand, as an estimate of the demand for service at any center, two different figures are typically used. The first one is the total population allocated to the center, multiplied by some experimental or practical factor, which gives an estimate of the expected number of simultaneous requests for service, or demanded workload, of that center. The second figure is an average of the historical rate of requests originating at the population allocated to the center, if a record of it is available. This demand is then constrained to be smaller than the maximum capacity of the center.

In the traditional optimization models described, the demand is implicitly assumed constant in time, equal to an average, which is strictly constrained to be smaller than the capacity of a center all the time. There is thus a contradiction between the strong, rigorous constraint, and the fact that it is applied to an average which, by definition, is exceeded 50% of the time. In this paper, we propose some models based on the fact that, in real life, the number of requests for service is not constant in time, but instead, it is a stochastic process. This stochasticity of the demand is explicitly taken into account in order to derive a capacity-like constraint which, instead of upper-bounding the demand to the capacity of the center, forces

a lower bound on the quality of the service at the center, particularly the waiting time or the number of people in line, awaiting for service.

We present several optimization models. Instead of minimizing the average response time, as other authors do, we restrict either the response time or the queue length to be smaller than a predetermined figure. This formulation allows us to obtain linear objectives and linear constraints, and thus, we are able to solve the location problem to optimality, if wanted. This formulation does not require of any approximations of the queueing model, as opposed to other location models in the literature.

A distinct contribution of the formulations presented here, as opposed to models that utilize averages, or models which minimize some quality indicator, is that *a particular value of service quality of the system* (as given by queue lengths or waiting times) *is explicitly embedded in the optimization model*. Thus, by using these models when designing a system, the resulting system can be fine-tuned, allowing the designer to trade off investment and operating cost versus service quality.

The first model addresses the issue of the location of the least number of single-server centers so that all the population is served within a standard distance, and nobody stands on line for a time longer than a given time-limit, or with more than a predetermined number of other clients. We then formulate several maximal coverage models, with one or more servers per service center.

The Queueing Maximal Covering Location-Allocation Model (QM-CLAM)

The maximal covering location-allocation model can be stated as:

"Locate *p* centers and allocate users to them so to maximize covered population, where coverage is defined as: i) covered population is allocated to a center within a time or distance standard from their home location, and ii) if a user is covered, at his/her arrival to the center, he/she will wait on a line with no more than *b* other people, with a probability of at least alpha."

or

"Locate *p* centers and allocate users to them so to maximize covered population, where coverage is defined as: i) covered population is allocated to a center within a time or distance standard from their home location, and ii) if a user is covered, at his/her arrival to the center, he/she will be served within time of his/her arrival to the center, with a probability of at least alpha."

The usual Church and ReVelle (1974) MCLP model can not be modified to accommodate the congestion constraint. Since there are no allocation variables in MCLP, it is not possible to allocate servers to demands covered by them. Without allocation variables, it is not possible for the model to aggregate all the calls for service arriving to a server and, as a consequence, to determine when congestion occurs. Thus, we rewrite the maximal covering model as a p-median-like model, using location and allocation variables.

The formulation of the model is the following:

$$\text{Max} \quad \sum_{i,j} a_i x_{ij} \qquad (1)$$

$$\text{s. t.} \quad x_{ij} \le y_j \qquad \forall i, j \qquad (2)$$

$$\sum_{j} x_{ij} \le 1 \qquad \forall\, i \qquad (3)$$

$$\text{P[center } j \text{ has} \le b \text{ people on queue]} \ge \alpha \qquad \forall j \qquad (4)$$

P[waiting time at center $j \leq$] $\geq \alpha$ $\forall j$ (4')

$$\sum_i y_i = p \qquad (5)$$

$x_{ij} = 0,1,\ j \in N_i.$ $\forall i,j$

$y_j = 0,1$ $\forall i,j$

Zero - one variable y_j is one if a center is located at node j, and zero otherwise. Zero - one variable x_{ij} is one if users located at demand node i are allocated to a center located at node j, and zero otherwise. If we define the set N_i either as the set of candidate locations that are within a distance standard from node i, or the set of candidate locations which can be reached from node i within a certain standard time, variables x_{ij} are only defined for the pair of subscripts (i, j) such that $j \in N_i$. This is because we are forcing coverage of every demand from a center located within the distance standard, hence there is no need to define variables that will never be equal to one. The parameter a_i is the total population at demand node i. Objective (1) maximizes the population allocated to a center. Note that there is no constraint forcing every demand node to be covered. Constraint (2) states that it is not possible to allocate a demand node i to a node j, unless there is a center at the last one. Constraint (3) forces each demand node i to be allocated to at most one service center j. While constraint (5) sets the number of centers to be located, constraint (4) forces every center to have less than, or at most, b people on line with a probability of at least α. This constraint assures that, on his/her arrival to the center, every user will find a line that is not longer than b, most of the time. Constraint (4') explicitly makes the total time spent by a user at the center shorter than or equal to with probability of at least α, assuring to every user a timely attention. Less tight constraints would be the usual capacity constraints:

Average, or expected Nr of customers in center $j \leq b$ $\forall j$ (4'')

Average, or expected time spent at center $j \leq$ $\forall j$ (4''')

The first one (4'') forces the *expected* number of users at center *j* (the one being attended plus those in queue) to be less than or equal to *b*. If this constraint holds, on their arrival to the center, users will find queues shorter than *b* a 50% of the time (not a $100*\alpha$ percent, as in constraint (4), and the other 50% of the time, they will face queues that are longer than *b*. Constraint (4'''), finally, forces the *average* time spent in the system to be lesser than or equal to .

In order to write constraint (4) in a tractable form, we make the reasonable (and customary) assumption that requests for service at each demand node *i* appear according to a Poisson process with intensity f_i. Since each center serves a set of demand nodes, the requests for service at that center are the union of the requests for service of the nodes in the set, and they can be described as another stochastic process, equal to the sum of several Poisson processes. This stochastic process can be easily shown to be also a Poisson process, with an intensity $_j$ equal to the sum of the intensities of the processes at the nodes served by the center. This set of nodes is not known before the solution of the mathematical programming problem is known. However, we can use variables x_{ij} in order to rewrite the parameter $_j$ as

$$_j = \sum_{i \in I} f_i x_{ij}$$

Using this definition, if a particular variable x_{ij} is one, meaning that node *i* is allocated to center *j*, the corresponding intensity f_i will be included in the computation of $_j$.

We also assume an exponentially distributed service time, with an average rate of $_j$ ($_j \geq _j$, otherwise the system does not reach an equilibrium). This is a reasonable assumption, since the travel time of clients is not considered, but only the service time at the center. If we assume steady state, we can use the well known results for a M/M/1 queueing system for each center and its allocated users.

If we define the state *k* of the system as *k* users in the system (either being attended or in queue), the state transition diagram of the system is the one shown in Figure 1.

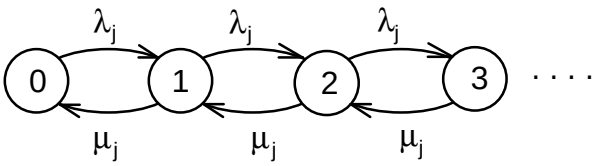


Figure 1. State transition diagram

In figure 1, state *k* corresponds to *k* users at the center, that is, state zero corresponds to the center being idle, state 1 to one user being attended at the center, state 2 to two users at the center: one of them getting attention and one in queue, and so on. We want to make the probability of a user being on a line with no more than *b* other people, at least equal to alpha. If we represent as p_k the steady state probability of being in state *k*, this requirement is written as:

$$p_0 + p_1 + ... + p_{b+1} \geq \alpha \qquad (6)$$

Writing and solving the steady state balance equations of the M/M/1 system, we get the following expression for the steady state probabilities [Wolff, 1989]:

$$p_k = (1 - \rho_j)\rho_j^k.$$

Where $\rho_j = {}_j / {}_j$. Hence, equation (6) becomes:

$$(1 - \rho_j) + (1 - \rho_j)\rho_j + (1 - \rho_j)\rho_j^2 + ... + (1 - \rho_j)\rho_j^{b+1} \geq \alpha ,$$

or

$$(1 - {}_j)\sum_{k=0}^{b+1} {}_j^k \geq \quad ,$$

which is equivalent to

$$(1- \quad _j)\frac{1- \quad _j^{b+2}}{1- \quad _j} \geq \qquad ,$$

or

$$1- \quad _j^{b+2} \geq \quad ,$$

$$_j^{b+2} \leq 1- \quad ,$$

$$_j \leq \sqrt[b+2]{1- \quad} .$$

Since $\rho_j = \quad _j / \quad _j$,

$$_j \leq \quad _j \sqrt[b+2]{1- \quad} . \qquad (7)$$

Equation (7) is equivalent to constraint (4). Using the relationship between the intensity at the center and the intensities at the demand nodes, constraint (4) is rewritten as

$$\sum_{i \in I} f_i x_{ij} \leq \quad _j \sqrt[b+2]{1- \quad} , \qquad (8)$$

which is a linear, deterministic equivalent of constraint (4).

If we choose to use constraint (4') instead of constraint (4), that is

$$\text{P[time spent at center } j \leq \quad] \geq \alpha \qquad \forall j ,$$

we may use the probability distribution function of the waiting time in a M/M/1 queue, *w*, which has the following expression (Larson, Odoni, 1981):

$$f_w(w_j) = (\quad_j - \quad_j) e^{-(\quad_j - \quad_j) w_j}$$

to derive its cumulative distribution:

$$P(w_j \leq \quad) = F_w(\quad) = 1 - e^{-(\quad_j - \quad_j)} \; . \qquad (9)$$

The probability in equation (9) is made greater than or equal to alpha:

$$1 - e^{-(\quad_j - \quad_j)} \geq \qquad \forall j,$$

$$e^{-(\quad_j - \quad_j)} \leq 1 - \qquad \forall j,$$

$$-(\quad_j - \quad_j) \leq \ln(1 - \quad) \qquad \forall j,$$

$$\quad_j \leq \quad_j + \frac{1}{-} \ln(1 - \quad) \qquad \forall j.$$

Rewriting the parameter $\quad_j$ as a function of the variables, we finally get

$$\sum_{i \in I} f_i x_{ij} \leq \quad_j + \frac{1}{-} \ln(1 - \quad) \qquad \forall j. \qquad (10)$$

Equation (10) is the linear, deterministic equivalent of constraint (4').

The final formulation of the model is the following:

$$\text{Max} \quad \sum_{i,j} a_i x_{ij}$$

$$\text{s. t.} \quad x_{ij} \leq y_j$$

$$\sum_j x_{ij} \leq 1 \qquad \forall\, i$$

$$\sum_i y_i = p$$

$$\sum_{i \in I} f_i x_{ij} \leq \quad _j \sqrt[b+2]{1-\quad} \ ,$$

or $$\sum_{i \in I} f_i x_{ij} \leq \quad _j + \frac{1}{-} \ln(1- \quad) \qquad \forall j.$$

$$x_{ij} = 0,1, \ j \in N_i. \qquad \forall i,j$$

$$y_j = 0,1 \qquad \forall i,j$$

If the problem is solved using its linear relaxation, in these last equations the right hand side may be multiplied by variable y_j in order to improve the integer characteristics of the variables at the solution.

The Queueing Maximal Covering Location-Allocation Model with co-location of *m* servers per center

This queueing maximal covering location model can be stated as:

"Locate *p* service centers, each with *m* servers (where *m* may depend on the location, that is, it could be an m_j) and allocate users to them so to maximize covered population, where coverage is defined as: i) covered population is allocated to a center within a time or distance standard from its home location, and ii) if a user is covered, at his/her arrival to the center, he/she will wait on a line with no more than *b* other people, with a probability of at least alpha."

The formulation of the model is the same as the previous one. We develop this model using only the constraint on queue length. Elsewhere, we formulate a model with an upper probabilistic bound on the waiting time. Again, we assume an exponentially distributed service time for each center, with a service rate of $_j$. If the number of servers at the center is m_j, the inequality $_j \geq m_j \ _j$ must hold; otherwise the system does not reach an equilibrium. By virtue of the above

assumptions, the well known results for a M/M/*m* queueing system can be used for each center and its allocated users.

The state transition diagram of the system is the one shown in Figure 2.

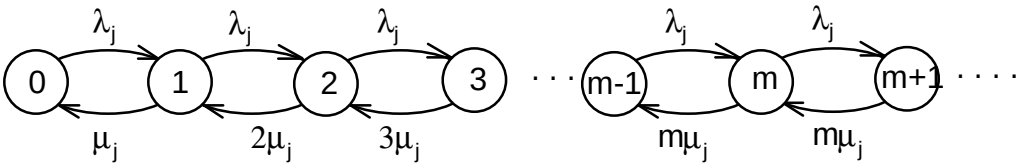


Figure 2. State transition diagram

In this figure, state *k* corresponds to *k* users at the center, that is, state zero corresponds to the center being idle, state 1 to one user being attended at the center, state 2 to two users at the center, both of them getting attention, and so on, up to state m, in which all m users in the system are getting attention. In state m + 1, however, m users are being attended and one in queue; state m + 2 represents m users in service and two in queue, and so on. We want to make the probability of a user finding a line with no more than *b* other people, at least equal to alpha. This requirement is written as:

$$p_0 + p_1 + ... + p_{m+b} \geq \alpha$$

that is, the probability of the queue being shorter than or equal to *b* users at the arrival of the next request, is greater than α. Also, since $p_0 + p_1 + ... + p = 1$,

$$p_{m+b+1} + p_{m+b+2} + ... + p \leq 1 - \alpha \qquad (11)$$

which means that the probability of the queue being longer than *b* is smaller than $1 - \alpha$. Note that the special case $b = 0$ does not mean that the user necessarily finds one server available, because it may happen that all *m* servers at the center are busy, but there are no users in queue. In this case, the arriving customer must wait

until one of the servers becomes idle. If immediate attention is desired, that is, at least one server free with probability α, then $p_0 + p_1 + ... + p_{m-1}$ must be forced to be greater or equal to α.

Solving the steady state balance equations of the M/M/*m* system, we get the following expression for the steady state probabilities [Wolff, 1989]:

$$p_k = p_0 \rho^k / k! \qquad k \le m$$

$$p_k = p_0 \rho^k / m!\, m^{k-m} \qquad k > m$$

$$p_0 = \left[\frac{m}{(1-\frac{}{m})m!} + \sum_{j=0}^{m-1} \frac{j}{j!} \right]^{-1}$$

Where $\rho =$ / . Although these parameters are specific to each server center, we will not use any subscript for the time being. With these expressions for the steady state probabilities, equation (11) becomes:

$$\sum_{k=m+b+1}^{k=\infty} \frac{p_0 m^m}{m!} \left(\frac{}{m}\right)^k \le 1 - \quad ,$$

or

$$\frac{p_0 m^m}{m!} \left(\frac{}{m}\right)^{m+b+1} \sum_{k=0}^{\infty} \left(\frac{}{m}\right)^k \le 1 - \quad .$$

Since $\rho/m \le 1$, the summation converges, and it can be written in a well known, simpler form. If we replace also the expression for p_0 , we get:

$$\left[\frac{m}{(1-\frac{}{m})m!} + \sum_{j=0}^{m-1} \frac{j}{j!} \right]^{-1} \frac{m^m}{m!} \left(\frac{\left(\frac{}{m}\right)^{m+b+1}}{1-\frac{}{m}} \right) \le 1 - \quad .$$

after some algebraic manipulation, this equation becomes

$$\sum_{k=0}^{m-1} \frac{(m-k)m!m^b}{k!} \frac{1}{m+b+1-k} \geq \frac{1}{1-} \qquad (12)$$

Since $\rho = \lambda/\mu$, and since λ is a function of the variables x_{ij}, equation (12) can be also written as a function of variables x_{ij}, becoming the deterministic equivalent of equation (4) for this case.

It is intuitively easy to see that, for any fixed value of α, the value of the left hand side of equation (12) can be made large enough to make the equation hold, by making ρ small enough, because its exponent is always positive. The value of variable ρ is decreased by manipulating variables x_{ij}, (making as many of them equal to zero as needed). Furthermore, for any value of α there must exist a value ρ of ρ which makes equation (12) hold as an equality, as well as a range of values of ρ such that equation (12) holds as a strict inequality.

Although it is the deterministic equivalent of equation (4), equation (12) can not be used in a linear model, because of its nonlinearity. However, we show next that its left hand side (LHS) is strictly decreasing with increasing ρ, and we later use this characteristic to find a linear equivalent to it.

Lemma: The left hand side of equation (12) strictly decreases with ρ.

Proof: The derivative of the LHS of (12) with respect to ρ is

$$\frac{LHS}{} = \sum_{k=0}^{m-1} \left[-(m+b+1-k)\right] \frac{(m-k)m!m^b}{k!} \frac{1}{m+b+2-k}. \qquad (13)$$

This derivative is strictly negative, because (m + b + 1 - k) is strictly positive, as well as all the remaining factors of each term of the summation. The entire summation is also strictly negative, and so is the whole derivative. Thus, the LHS of equation (12) is a strictly decreasing function of ρ, which also means that it strictly increases when ρ decreases ■.

Theorem: Since the left hand side of equation (12) is strictly decreasing with ρ, if ρ^* is the value of ρ which makes equation (12) hold as an equality, the equation:

$$\rho \leq \rho^* \quad ,$$

is a linear equivalent to (12), when used as a constraint for a linear programming formulation.

Proof: Let ρ^* be the value of ρ which makes equation (12) hold as an equality. Since the LHS strictly increases when ρ decreases, for any value of $\rho \leq \rho^*$, equation (12) also holds. In other words, the inequality $\rho \leq \rho^*$ is a sufficient condition for equation (12) to hold. Furthermore, by virtue of the strictly increasing nature of the LHS of equation (12), for any $\rho > \rho^*$ equation (12) can not hold. Thus, $\rho \leq \rho^*$ is also a necessary condition for this equation to hold. Since $\rho \leq \rho^*$ is a necessary and sufficient condition for equation (12) to hold, we can use it instead of this equation■.

Once the value of α is given, the value of ρ^* can be found by using any numeric root - finding technique (Newton methods, for example) on equation (12), written as an equality, and equation

$$\rho \leq \rho^* \qquad (14)$$

is the new deterministic, linear equivalent to equation (12). This procedure must be repeated for each service center *j,* and a value ρ_j found for each one, obtaining the set of equations

$$\rho_j \leq \rho_j \qquad \forall j.$$

Since $\rho_j = \lambda_j / \mu_j$,

$$\lambda_j \leq \mu_j \rho_j \qquad \forall j.$$

Recalling that λ_j is a function of variables x_{ij},

$$\sum_{i \in I} f_i x_{ij} \leq \mu_j \rho_j, \qquad \forall j \qquad (15)$$

which is the set of linear, deterministic equivalents of constraint (4). Remember also, that μ_j is a function of the number of servers at center *j* (m_j) which can be specified for each center before solving the model. The final model is:

$$\begin{aligned}
\text{Max} \quad & \sum_{i,j} a_i x_{ij} \\
\text{s. t.} \quad & x_{ij} \leq y_j \\
& \sum_j x_{ij} \leq 1 && \forall i \\
& \sum_i y_i = p \\
& \sum_{i \in I} f_i x_{ij} \leq \mu_j \rho_j, && \forall j \\
& x_{ij} = 0,1,\ j \in N_i. && \forall i,j \\
& y_j = 0,1 && \forall i,j
\end{aligned}$$

In this model, variable x_{ij} is one if m_j servers are located at center j, and zero otherwise.

Computational Experience: Branch and Bound and Heuristic Methods

We assumed that the service centers are primary health care centers. The servers are physicians, and there is one or more physicians at each center. The average service time was set at 20 minutes. We used the 30 node network, shown in Figure 3. Each demand center is also a potential server location, and the distances are Euclidean. We assumed that centers serve only patients located at a distance of 1.5 miles or less.

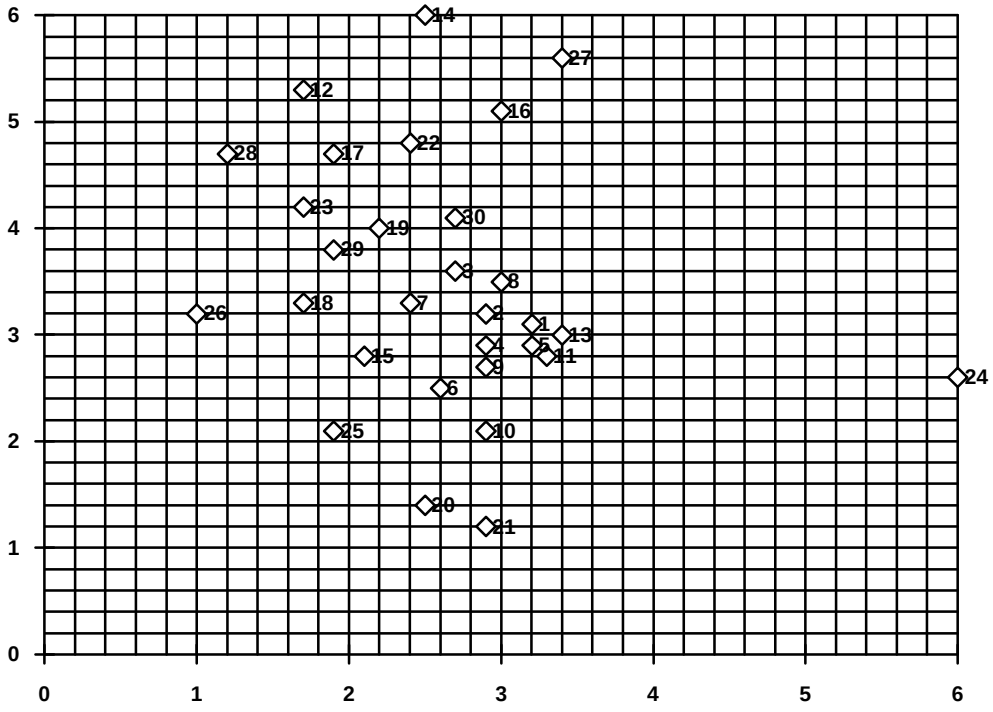


Figure 3. 30-node network

The populations and coordinates of the nodes of the network are shown in Table 1. Assuming one server per center, we computed the value of the right hand side of equation (8) (limit values of arrival rates, λ), given an average service time $1/\mu$ of 20 minutes, different values of α, and different values of queue length b. In

order to compare both criteria of quality of service (queue length and total response time), we also show the total response time *w* (time in queue plus time in service which the clients will not exceed with probability α), that would result in the same values of λ if equation (10) is used instead of equation (8). The values are shown in Table 2.

Queue length 0 means that, at the time a new user arrives, there are either no other users in the system, or there is one user being attended by the server. It is interesting to note that, for a particular value of λ, total response time appears intuitively to be very long as compared to queue length. This is due to the fact that, when computing a queue length of, say, 5, only occupation states zero to six (no users in the system, up to one user being attended plus five users on line), with their associated probabilities p_0 to p_6 have to be considered. However, when dealing with response time, the probabilities of long waiting times have to be aggregated over **all** possible states *k*.

Branch and Bound. Constrained waiting time. One server per center

The commercial Integer Programming software CPLEX 3.0, on a cluster of eight DEC 3000 - 700 AXP computers was used to solve the QM-CLAM model. The branching was limited to 20000 in each case. With this limit, the run time was between 10 and 40 minutes per run. The daily call rate for each demand node was set at 0.006 times the population.

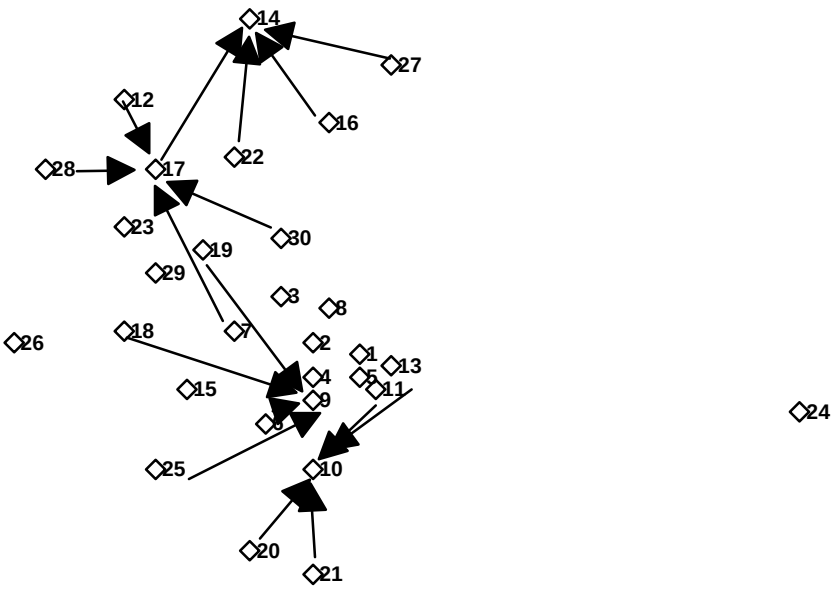
Table 3 shows the results for α = 85%, 90% and 95%, and different values of the waiting time, τ. In this table, S is the number of servers to be located. a "y" in the columns "L" indicate when the branching limit of 20000 was reached. The columns "Obj" indicate the value of the objective and the columns "Locns" the locations of the centers. Note that, for short waiting times, a difference of one minute can change dramatically the coverage. This can be seen for example in the cases τ =

48 and 49, α = 90%. For τ = 49 min, 9 servers are enough for complete coverage of the population. However, if the required waiting time is reduced in one minute, complete coverage becomes impossible, no matter how many servers are deployed.

The limit of 20000 branches was reached in many of the runs. In these cases, the solution is not necessarily optimal. Note also, that there are some cases in which no matter how many servers are located in the system, it is not possible to achieve total coverage of the population, and some demand nodes are not covered by the service. This is due to the fact that the call rate of these nodes exceeds the service capability of a single server.

Given its capacity-like form, when the probabilistic constraint was binding, the run time and the number of branches was much higher than in the case in which the distance limit was binding.

Since allocation to nearest server was not required, in many cases a demand was allocated to a server that is not the nearest. In practice, instead of choosing a server because of its closeness, users go to a server which gives a better service in terms of waiting time. Figure 4 shows one of these cases. If allocation to nearest server is desired, extra constraints have to be included in the models.



F

iFigure 4. CPLEX solution for τ = 48, α = 90%. The 4 servers are located at nodes 9,10,14 and 17. The arrows show the allocations.

Branch and Bound. Constrained queue length. One or more servers per center

The queue lengths were set at 0, 1, and 2 users. In order to show a more congested case, The call rates for one server per center were set at 0.015 times the node population. For 3 and 5 servers per center, the call rates were set at 0.042 times the node population. These last cases represents a highly congested system.

The results for 1 server are shown in Table 4 for three values of alpha: 85%, 90% and 95%. The results for 3 and 5 servers are shown in Table 5, only for α = 95%. Again, the branching limit was reached in many cases, specially for 3 and 5 servers per center, suggesting solutions that are not necessarily optimal.

Several simple heuristics were developed for obtaining fast solutions of the models. These heuristics were all based on satisfying first the time limit or queue length restriction. We only describe the best one, which follows:

Step 0: - Make a list of the candidate nodes *j* (Ordered by decreasing own call rate f_j, if they are also demand nodes. Ordered at random if the candidate nodes are not demand nodes). Call this list D_j.

- Compute, for each candidate node, the right hand side of equation (8) or (10), depending on what model is being utilized. This right hand side is the limit value of average calls per time unit ($\lambda_{lim,j}$) that can be accepted by a center located at this node. If this limit value is exceeded, the constraint on queue length or waiting time limit does not hold.

- For each candidate node *j*, make a list of all the demand nodes *i* within distance standard, ordered by increasing distance to the candidate node. Call this list D_{ji}. Note that the same demand node can be in several lists D_{ji}.

Step 1 For each candidate node, make the current total incoming call rate equal to zero ($\lambda_{inc,j}$).

Step 2 Starting with the first candidate node *j* on the list D_j, add to its incoming call rate $\lambda_{inc,j}$, the call rate f_i of the first node on the list D_{ji}. Then, add the call rate f_i of the second node on the list D_{ji}, then the third, and so on, until the point in which the adding of any extra demand node would exceed the limit value of calls $\lambda_{lim,j}$. Allocate temporarily all these demand nodes to a hypothetical server at node *j*.

Step 3 Repeat Step 2 for all nodes in list D_j. Note that the same demand node could be temporarily allocated to several candidate locations.

Step 4 Locate a server on the node with the highest $\lambda_{inc,j}$. Take all demand nodes allocated to it, out of all the lists D_{ji}, of all potential centers. Allocate them definitively to the located server.

Step 5 Take each one of the servers already located, de-allocate the demands that were allocated to it, and move it to all possible unused candidate locations (repeating each time steps 1 to 4). If some location gives a better objective, keep the server at that location. Repeat for all servers already located. If there is any improvement on the solution, repeat step five until there are no further changes that improve the solution.

Step 6 Repeat steps 1 to 5 until all available servers are located.

Step 7 Repeat procedure of step 5, for all servers.

Two versions of the heuristic were run for each value of the parameters. The first one, with step 7, but without step 5, and the second one, with steps 5 and 7. For all cases, the heuristic took no longer than 2.5 seconds, on a 486SX computer, running at 100 MHz. The second version performed better in most cases, and we show only results of the second heuristic.

Heuristics. Constrained waiting time. One server per center

The following tables show the results obtained by the heuristic. The figures shown correspond to the average percentage of difference between heuristic solution and CPLEX solution, for different numbers of servers. Also shown are the maximum percentages of difference between both solution methods.

It is interesting to note that, for α=95%, τ = 62 m, the solution given by the heuristic is better than the CPLEX solution. This happens in many cases, and it is due to the branching limit imposed on CPLEX. Because of this branching limit, the solution found by CPLEX is not optimal, and the heuristic finds a better solution than the Integer Programming package.

For one server, constrained waiting time, the run time for both heuristics never exceeded 0.11 seconds.

Due to the structure of the heuristics, it is robust in many cases, in the sense that when more than *k* servers are being located, the locations of the first *k* servers correspond to the best solution to the problem of locating *k* servers. In other words, the solution (locations) to the problem of locating *k* servers is a subset of the solutions to the problems of locating more than *k* servers. When CPLEX is used, however, locations may be very different for different numbers of servers, possibly because different CPLEX runs choose different alternate optima.

Because of how the heuristics fill the service capacity of each potential center, in most of the cases the resulting "districting" consists of attention regions that do not overlap, as opposed to the regions found by CPLEX. For example, figure 5 shows the heuristic solution for τ = 48, α = 90%. The CPLEX solution for the same case, displayed in Figure 4, shows a center located on a demand node (17) which is assigned to a different center (14). Also, it shows attention regions that overlap, as in the case of demand nodes 7, 18 and 19.

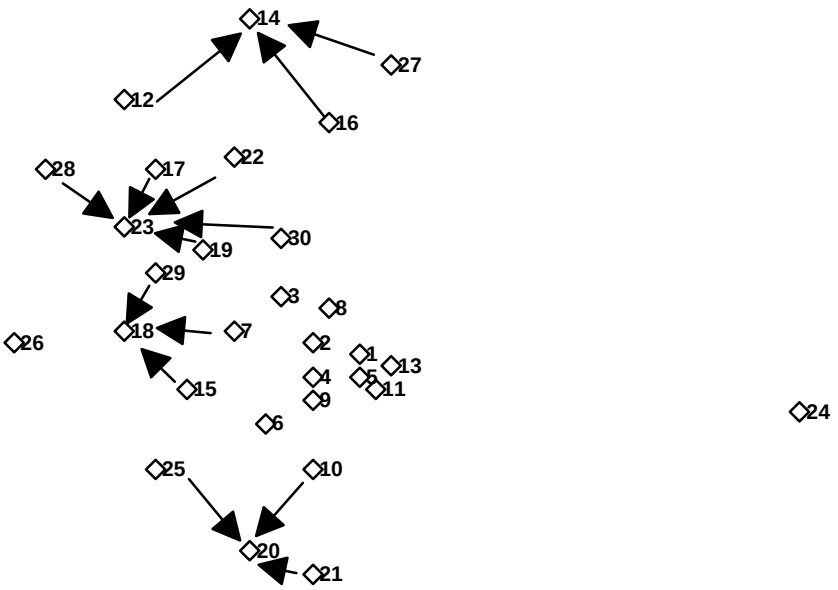


Figure 5. Heuristic solution for $\tau = 48$, $\alpha = 90\%$. The 4 servers are located at nodes 16,18,20 and 23. The demand at these is self-assigned. The arrows show the allocations.

Heuristics. Constrained queue length. One or more servers per center

Table 7 shows the results of the heuristic, compared to the solutions obtained with CPLEX. The column Ohe displays the value of the objective, while the column "%" shows the percentage of difference between CPLEX and the heuristic. Table 8 shows the results for 3 and 5 servers located at each center, for $\alpha = 95\%$.

Conclusions

New models for locating service centers in a congested situation have been presented. These models, being linear, include explicitly a constraint on service quality, specifically the waiting time or queue length at each center. Heuristics solutions are presented for the models, and compared to solutions obtained with a commercial integer programming package (CPLEX). Heuristic solutions appear to be very close to the solutions obtained by using CPLEX, when the run time (or

branching) is limited. In some cases, heuristic solutions are better. These heuristics are very simple, and can be improved introducing random-selection steps and/or random restarts in them. Further improvements in the objective value could be expected if a knapsack problem is solved heuristically each time the allocations are made.

Node	X	Y	Popn
1	3.2	3.1	710
2	2.9	3.2	620
3	2.7	3.6	560
4	2.9	2.9	390
5	3.2	2.9	350
6	2.6	2.5	210
7	2.4	3.3	200
8	3.0	3.5	190
9	2.9	2.7	170
10	2.9	2.1	170
11	3.3	2.8	160
12	1.7	5.3	150
13	3.4	3.0	140
14	2.5	6.0	120
15	2.1	2.8	120
Node	**X**	**Y**	**Popn**
16	3.0	5.1	110
17	1.9	4.7	100
18	1.7	3.3	100
19	2.2	4.0	90
20	2.5	1.4	90
21	2.9	1.2	90
22	2.4	4.8	80
23	1.7	4.2	80
24	6.0	2.6	80
25	1.9	2.1	80
26	1.0	3.2	70
27	3.4	5.6	60
28	1.2	4.7	60
29	1.9	3.8	60
30	2.7	4.1	60

Table 1: test network

$\alpha = 0.99$:

b	λ *[1/min]*	*w* *[min]*
0	0.005	102.33
1	0.0108	117.35
2	0.0158	134.70
3	0.0199	153.10
4	0.0232	171.90

$\alpha = 0.9$:

b	λ *[1/min]*	*w* *[min]*
0	0.0158	67.35
1	0.0232	85.94
2	0.0281	105.22
3	0.0315	124.79
4	0.0341	144.50

$\alpha = 0.5$:

b	λ *[1/min]*	*w* *[min]*
0	0.0354	47.5
1	0.0397	67.5
2	0.042	87.5
3	0.0435	107.0
4	0.0445	127.0

Table 2: Limit values of arrival rates

α = 85%

S	τ=40			τ=41			τ=42			τ=49			τ=52		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
9	4140	n	1,3,6,7,13, 16,23,24,30	-		-	-	-	-	-	-	-	-	-	-
8	4140	n	1,2,6,14,19, 24,25,30	5470	n	1,6,7,8,9, 19,22,24	-	-	-	-	-	-	-	-	-
7	4060	y	2,4,7,17, 22,25,30	5390	y	1,3,4,5, 10,22,25	-	-	-	-	-	-	-	-	-
6	3500	y	1,7,10, 13,15,22	5110	y	2,4,6,7,17, 22	5470	n	5,6,7,15,22, 24	-	-	-	-	-	-
5	2940	y	5,6,7,16,19	4310	y	4,7,10,22,30	5390	y	1,7,9,15,22	-	-	-	-	-	-
4	2370	y	6,9,19,22	3500	y	6,22,29,30	4520	y	7,10,15,22	5470	n	10,15,22,24	5470	n	10,15,22,24
3	1830	y	4,19,22	2670	y	2,19,29	3400	y	9,10,17	5390	n	6,15,22	5400	n	6,22,24
2	1220	y	9,19	1780	y	4,7	2300	y	6,29	5210	n	9,19	5320	n	9,22

α = 90%

S	τ=48			τ=49			τ=50			τ=60			τ=70		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
9	3580	n	5,6,8,10,14, 23,24,26,30	5470	n	1,2,3,6,7,18, 22,24,30	-	-	-	-	-	-	-	-	-
8	3500	y	4,6,9,10,14, 15,23,30	5390	y	3,7,9,11,13, 15,22,30	-	-	-	-	-	-	-	-	-
7	2960	y	4,6,7,10,12, 18,27	4880	y	1,4,6,7,8,9, 22	5470	n	1,10,11,19, 22,24,29	-	-	-	-	-	-
6	2710	y	5,6,7,17,18, 22	4240	y	1,7,15,19, 22,30	5390	y	6,7,9,15,22, 30	-	-	-	-	-	-
5	2300	y	3,6,14,15,17	3530	y	9,10,17,19,22	4580	y	3,6,7,22,29	-	-	-	-	-	-
4	1900	y	9,10,14,17	2750	y	8,17,19,25	3720	y	10,11,15,19	5470	n	9,22,24,29	5470	n	15,20,22,24
3	1430	y	15,19,22	2140	y	9,17,25	2810	y	1,5,25	5390	n	6,15,22	5400	n	9,22,24
2				1430	y	1,7	1880	y	9,22	5210	n	9,19	5320	n	9,22

α = 95%

S	τ=62			τ=63			τ=64			τ=74			τ=84		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
11	3580	n	1,2,3,9,10,12,13, 16,24,25,30				-	-	-	-	-				
10	3520	y	1,2,3,9,10, 12,13,16,24,25	4140	n	-	-	-	-	-	-				
9	3340	y	3,8,9,10,11,12,13, 25,27	4140	n	2,5,13,15,17, 22,25,29	5470	n	1,2,6,9,13,1 5,22,24,29	-	-	-	-	-	-
8	2970	y	3,5,10,12,13,14,20 ,25	4060	y	2,6,7,8,14, 15,17,29	5390	y	1,2,6,9,13, 22,25,29	-	-	-	-	-	-
7	2670	y	5,6,7,10,12,14,25	3940	y	1,3,6,7,15, 22,25	5180	y	6,7,8,9,17, 22,30	-	-	-	-	-	-
6	2280	y	2,4,9,12,14,25	3300	y	4,7,10,15, 19,22	4350	y	2,7,9,10,15, 19	-	-	-	-	-	-
5	1900	y	14,15,16,23,25	2730	y	10,12,15, 22,25	3680	y	4,9,10,19,2 9	-	-	-	-	-	-
4	1570	y	9,13,14,19	2270	y	7,10,22,30	2890	y	9.10.29,30	5470	n	7,10,22,24	5470	n	6,15,22,24
3				1720	y	22,25,30	2260	y	5,9,26	5390	n	7,10,22	5400	n	6,22,24
2				1160	y	7,16	1520	y	20,23	5210	y	2,6	5320	n	6,22

Table 3: CPLEX results for 1 server per center.

α = 95%

S	b=0			b=1			b=2		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
7	5470	n	1,2,5,6,15,22,24	-	-	-	-	-	-
6	5390	y	2,10,11,20,22,23	-	-	-	-	-	-
5	5260	y	1,6,7,22,30	5470	n	2,9,15,22,24	-	-	-
4	4170	y	7,15,17,25	5390	y	1,6,15,22	5470	n	6,15,22,24
3	3170	y	9,10,25	5270	y	9,15,17	5390	n	7,9,22
2	2140	y	6,19	3520	y	10,18	4540	y	8,19

α = 90%

S	b=0			b=1			b=2		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
5	5470	n	3,7,10,22,24	-	-	-	-	-	-
4	5390	n	6,7,22,25	5470	n	6,15,22,24	5470	n	10,15,22,24
3	4480	y	7,10,22	5390	n	10,19,22	5390	n	6,15,22
2	3020	y	10,17	4440	y	2,7	5210	n	9,19

α = 85%

S	b=0			b=1			b=2		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
4	5470	n	10,22,24,29	5470	n	6,15,22,24	5470	n	10,15,22,24
3	5390	n	6,7,22	5390	n	9,15,22	5390	n	6,15,22
2	3700	y	1,19	5100	n	6,19	5210	n	9,19

Table 4: CPLEX results for 1 server per center

3 servers per center, α = 95%

S	**b=0**			**b=1**			**b=2**		
	Obj	**L**	**Locns**	**Obj**	**L**	**Locns**	**Obj**	**L**	**Locns**
10	4760	n	1,2,3,5,6,11,14, 15,18,24						
9	4760	n	3,4,5,6,7,10,13, 22,24						
8	4680	y	3,5,7,9,15,19,22, 23	5470	n	2,7,8,9,15, 19,22,24			
7	4530	y	1,6,7,9,19,22,29	5390	y	4,6,8,9,15, 22,30	5470	n	10,13,15, 22,24,29,30
6	3860	y	6,9,17,22,25,30	4620	y	6,9,10,17, 19,30	5390	y	6,9,11,22, 29,30
5	3230	y	6,9,15,16,19	3920	y	6,7,10,13,19	4620	y	6,7,8,22,29
4	2490	y	9,10,17,22	3210	y	3,9,15,19	3630	y	2,10,15,19
3	1960	y	4,5,20	2460	y	15,19,25	2810	y	15,19,25
2	1320	y	15,28	1640	y	19,23	1880	y	9,22

5 servers per center, α = 95%

S	b=0			b=1			b=2		
	Obj	L	Locns	Obj	L	Locns	Obj	L	Locns
6	5470	n	6,10,14,15,24,29						
5	5410	y	9,10,22,24,29	5470	n	6,10,15,22,24	5470	n	1,9,15,22,24
4	5320	y	7,9,22,30	5390	n	6,10,15,22	5390	y	6,7,15,22
3	4010	y	7,11,23	4570	y	7,8,19	5070	y	6,15,22
2	2860	y	2,15	3080	y	9,22	3400	y	3,13

Table 5

α=85%

τ = 40 m	τ = 41 m	τ = 42 m	τ = 49 m	τ = 52 m	τ = 62 m
4.17	5.04	3.10	0.00	2.06	5.49
(7.64)	(9.39)	(6.12)	(0.00)	(13.16)	(13.16)

α=90%

τ = 48 m	τ = 49 m	τ = 50 m	τ = 60 m	τ = 70 m
3.98	1.21	3.15	1.66	3.16
(7.43)	(3.69)	(5.34)	(4.99)	(4.51)

α=95%

τ = 62 m	τ = 63 m	τ = 64 m	τ = 74 m	τ = 84 m
-0.42	1.62	0.78	7.48	2.74
(0.90)	(5.33)	(5.60)	(12.86)	(5.83)

Table 6: Results of the heuristics. The figures in parenthesis correspond to maximum values of percentage of difference between heuristic and CPLEX. The figures on top are the average percentages of difference between both solution methods.

α = 95%

S		b=0			b=1			b=2	
	%	Ohe	Locns	%	Ohe	Locns	%	Ohe	Locns
7	0	5470	6,11,18,1, 22,20,24	0	5470	2,4,17,20,24, 26,27	-	-	-
6	0	5390	6,11,18,1, 22,20	1.10	5410	2,4,17,20,24, 26	-	-	-
5	0.95	5210	6,11,18,1, 22	2.38	5340	2,4,17,20,24	0	5470	2,9,22,21,24
4	-1.68	4240	1,4,18,19	2.41	5260	2,4,17,20	1.28	5400	2,9,22,21
3	-0.31	3180	1,15,19	3.60	5080	2,4,17	2.97	5230	2,9,22,
2	0.94	2120	1, 6	0.28	3510	2,4	1.54	4470	2,9

α = 90%

S		b=0			b=1			b=2	
	%	Ohe	Locns	%	Ohe	Locns	%	Ohe	Locns
6	0	5470	7,18,1,22, 10,24	-	-	-	-	-	-
5	1.46	5390	7,18,1,22, 10	0	5470	15,3,22,25,24	-	-	-
4	6.49	5040	7,18,1,22	1.46	5390	15,3,22,25	0	5470	15,30,12,20
3	2.68	4360	7,18,1	6.12	5060	15,3,22	0	5390	15,30,12
2	0.67	3000	7,18	1.35	4380	15,3	0	5210	15,30

α = 85%

S		b=0			b=1			b=2	
	%	Ohe	Locns	%	Ohe	Locns	%	Ohe	Locns
5	0	5470	7,10,19,14, 24	0	5470	2,7,22,20,24	5470	0	4,19,10,14,24
4	1.46	5390	7,10,19,14	1.46	5390	2,7,22,20	5390	1.46	4,19,10,14
3	3.34	5210	7,10,19	3.34	5210	2,7,22	5210	3.34	4,19,10
2	0	3700	7,10	9.21	4630	2,7	4780	8.25	4,19

Table 7

3 servers per center, α = 95%

S		b=0			b=1			b=2	
	%	Ohe	Locns	%	Ohe	Locns	%	Ohe	Locns
10	0	4760	11,18,20,17,2,3,4,8,14,24						
9	1.68	4680	11,18,20,17,2,3,4,8,14	0	5470	1,2,5,12,15 20,24,27,29	0	5470	
8	3.85	4500	11,18,20,17,2,3,4,8	1.0	5410	1,2,5,12,15,20,24,29	1.10	5410	
7	4.86	4310	11,18,20,17,2,3,4	1.1	5330	3,15,20,2,4,1,22	2.56	5330	10,13,15,22,24,29,30
6	2.85	3750	11,18,20,17,2,3	-1.95	4710	3,15,20,2,4,1	4.45	5150	6,9,11,22,29,30
5	1.24	3190	11,18,20,17,2	-2.04	4000	3,15,20,2,4	3.68	4450	6,7,8,22,29
4	-3.21	2570	11,18,20,17	-1.56	3260	3,15,20,2	0.83	3600	2,10,15,19
3	1.53	1930	11,18,20	0.41	2440	3,15,20	2.85	2730	15,19,25
2	2.27	1290	11,18	1.2	1620	20,15	1.60	1850	9,22

5 servers per center, α = 95%

S		b=0			b=1			b=2	
	%	Ohe	Locns	%	Ohe	Locns	%	Ohe	Locns
6	0	5470	8,30,5,25,12,24	0	5470				
5	0.36	5390	1,8,9,19,22	1.46	5390	10,23,2,15,16	0	5470	1,9,15,22,24
4	6.01	5000	8,30,5,25	1.67	5300	10,23,2,15	0	5390	6,7,15,22
3	0.25	4000	8,10,18	-0.88	4610	10,23,15	2.37	4950	6,15,22
2	7.30	2650	19,25	0	3080	10,23	0.29	3390	3,13

Table 8

References

1. Batta R, 1988, "Single Server Queueing - Location Models with Rejection", *Transportation Science*, **22**, 209 - 216.

2. Batta R, 1989, "A Queueing - Location Model with Expected Service Time dependent Queueing Disciplines", *European Journal of Operational Research*, **39**, 192 - 205.

3. Batta R, Larson R, Odoni A, 1988, "A Single - Server Priority Queueing - Location Model", *Networks*, **8**, 87 - 103.

4. Berman O, Larson R, 1985, "Optimal 2 - Facility Network Districting in the Presence of Queueing", *Transportation Science*, **19**, 261 - 277.

5. Berman O, Larson R, Chiu S, 1985, "Optimal Server Location on a Network Operating as a M/G/1 Queue", *Operations Research*, **12(4)**, 746 - 771.

6. Berman O, Larson R, Parkan C, 1987, "The Stochastic Queue *p* - Median Location Problem", *Transportation Science*, **21**, 207 - 216.

7. Berman O, Mandowsky, R, 1986, "Location - Allocation on Congested Networks", *European Journal of Operational Research,* **26**, 238 - 250.

8. Church R. ReVelle C., 1974, "The Maximal Covering Location Problem", *Papers of the Regional science Association*, **32**, 101 - 118.

9. Hakimi S. L., 1964 "Optimal Locations of Switching Centers and the absolute Centers and Medians of a Graph", *Operations Research*, **12**, 450 - 459.

10. Larson R, Odoni A, 1981, *Urban Operations Research*, Prentice-Hall, Englewood Cliffs, NJ.

11. Marianov V, ReVelle C, 1994, "The Queueing Probabilistic Location Set Covering Problem and Some Extensions", *Socio-Economic Planning Sciences*, **28(3)**, 167 - 178.

12. ReVelle C. and Swain S., 1970, "Central Facilities Location", *Geographical Analysis*, **2**, 30 - 42.

13. Wolff R, 1989, *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, Englewood Cliffs, NJ.