HIERARCHICAL LOCATION - ALLOCATION MODELS FOR CONGESTED SYSTEMS

by

Vladimir Marianov[1]
Department of Electrical Engineering
The Catholic University of Chile, Santiago, CHILE
E-mail: *marianov@ing.puc.cl*

and

Daniel Serra
Department of Economics and Business, IET and CRES
Universitat Pompeu Fabra, Barcelona, SPAIN
E-mail: *daniel.serra@econ.upf.es*

VERSION: 24.01.2000

In this paper we address the issue of locating hierarchical facilities in the presence of congestion. Two hierarchical models are presented, where lower level servers attend requests first, and then, some of the served customers are referred to higher level servers. In the first model, the objective is to find the minimum number of servers and their locations that will cover a given region with a distance or time standard. The second model is cast as a Maximal Covering Location formulation. A heuristic procedure is then presented together with computational experience Finally, some extensions of these models that address other types of spatial configurations are offered.

[1] This research has been possible partly thanks to a grant by FONDECYT-Chile (Project Nr 1980815)

1. Introduction

Systems with a hierarchical structure are common both in public and private sectors. Examples of hierarchical structures can be found in the public health services, where hospitals correspond to the higher level facilities, and primary health care centers are the lower level. For example, U.S. school systems are hierarchical in nature, being composed by primary schools (kindergarten to fifth grade), middle schools (sixth through eight) and high schools (ninth through twelfth grade). In the telecommunications area, there are many examples of hierarchical networks; particularly, the star - star concentrator network is a system with different levels of servers, being the concentrators the lower hierarchy servers and the central node (or nodes), the higher level servers. Bank branches and automatic teller machines are yet another example of such hierarchical structures. In all these cases, the hierarchy can be generalized to more than two levels.

In hierarchical systems, facilities at different levels provide different types of services. However, there is often a linkage between the different levels, which makes impossible to solve the location problem for each level separately. For example, in a health service, all the customers of a particular primary health center are usually referred to the same hospital for high-level service. Especially if there are capacity constraints, this form of providing the services establishes a link between levels. When a high-level server provides also low-level services (as is often the case), the problem is obviously non-separable.

Hierarchical services can be classified according to their general structure (Narula, 1985). In a *nested* hierarchy, a high-level server provides also low-level service. In a *non-nested* hierarchy, servers of each level offer a different service. A *coherent* hierarchical system is one in which all customers of the same low-level server are also customers of the same high-level server. A *referral* system, as opposed to a *non-referral* system, is one in which users can not go to the higher level server unless a low-level server refers them to it. There have been several models for design (facility location and/or customer allocation to facilities) of hierarchical systems. A good review of the early models is given in Church and Eaton, (1987) and Gerrard and Church (1994). Later models include the PQ-median model, by Serra and ReVelle (1993, 1994), which combines hierarchical location and coherent districting, a hierarchical maximum capture model for location in a competitive environment, by Serra, Marianov and ReVelle (1992) and the Coherent Covering Location Problem, by Serra (1996).

In a different context, congestion is an issue that has attracted much attention. The reader can find the previous work in papers by Batta *et al*, 1988 and 1989, Batta *et al*, 1985, Berman *et al,* 1985, Berman *et al,* 1987, Berman and Mandowsky, 1986, Daskin, 1983, Batta *et al*, 1989, Larson and Odoni, 1981, Marianov and ReVelle, 1994 and 1996, Marianov and Serra, 1998. However, it has scarcely been addressed in the context of hierarchical models. A model by Mandell (1996) considers a hierarchical (two-tiered) emergency medical service in which two types of ambulances cover a population. Congestion happens when a service center is not capable to serve all the simultaneous requests for service that are made to it. Often, in order to avoid the effects of congestion, a capacity constraint is added to the design model, reflecting the fact that each server is capable of serving up to a certain number of requests before getting congested. This is a deterministic approach to the problem and, depending on how this capacity constraint is developed, the solution to the model is either a system with an unnecessarily large number of servers, or a system that is just not capable to attend all the demand. In other words, usual capacity constraints do not reflect the dynamic nature of congestion, in which time has a paramount role. This dynamic characteristic is explicitly taken into account in the two-level hierarchical models we propose, in which the capacity-like constraints are developed from an explicit probabilistic description of the system. Congestion is assumed to happen at both levels of the system, and as a consequence of it, requests for service are put in a queue, having to wait for some time before being served. In a public health system, for example, this would mean that people traveling to the primary health care center location, have to stand in line with more people, and when they are referred to the hospital, they have to wait again. In a star - star concentrator location problem (telecommunications network design problem), it means that at every concentrator there is a queue of messages, waiting for the access to the connection to central nodes, in which there are in turn other queues of messages waiting to be processed. In the models we present, we develop a constraint that sets a lower bound of α on the probability of a request being on a queue with at most *b* other requests, where the value of *b* can be different for each level. Quality of service could be enforced not only through constraints on the queue length, but also through constraints on the waiting time. We chose queue length because it is simpler, and because perceived quality of service depends mainly on the number of people that an arriving customer finds in line, at her/his arrival to the center. Furthermore, queue length is an acceptable proxy for waiting time, since there is a strong relation between both indicators.

Note that, although it could be argued that in certain cases travel time should be added to waiting time, in this work, we do not consider travel time. Adding both times would be appropriate in competitive environments, in which attractiveness of the centers is the goal of the planner. In this case, the different travel modalities to the centers should be also taken into consideration. The use of travel time would also be appropriate in emergency contexts, in which the sum of travel plus waiting time is a very important indicator of the effectiveness of the service. In this last case, the server travels to the place where a call appears. In our model, the customer travels to the center, and the formulation applies to what is called a dictatorial environment, in which assignments are made by an authority just having in mind to assure the best possible quality of service once the customers arrive to the centers. This is also the reason for not considering assignment to closest centers. Finally, considering travel time would complicate unnecessarily the models, since we chose to use queue length instead of waiting time.

Two types of models are presented in the remainder of the paper. We first present a Hierarchical Queuing Location Set Covering formulation, which seeks complete coverage of the population while minimizing the number of servers. Next, we formulate models that maximize the population covered when a limited number of servers is sited (Hierarchical Queuing Maximal Covering Location Models). Although the models we present in the following sections were developed for referral systems, they can easily be applied to non-referral systems. Also, the models can be applied to nested and to non-nested systems. In the case of nested systems, a server providing both high-level and low-level services is modeled as a low-level server co-located with a high-level server. In our formulations, allocation of each demand point to a server is unique; that is, a customer does not have the possibility of going to a different server when the queue at the assigned server is too long, for example (“dictatorial” environment).

After presenting the models, we offer some extensions related to the spatial organization of the services to be located. Finally, we propose a heuristic to solve the models, together with some computational experience.

2. HiQ-LSCP: Hierarchical Queuing Location Set Covering Problem

The Hierarchical Queuing Location Set Covering Problem can be stated as follows:

"Minimize the cost of locating low-level and high-level service centers in such a way that, while all demand for both levels of service must be served by centers located within a specified distance of its origin, the probability of a customer standing in a line with *b* other customers is at most α."

In other words, in the HiQ-LSCP model, full coverage of population is mandatory, while the number of servers at both levels is minimized. Complete coverage of all population in a demand node is attained when two conditions are met: First, the demand node is allocated to servers of both levels, located within specified standard distances from demand to low-level server, demand to high-level server and low-level server to its high-level server. We will call this the allocation condition. Secondly, a user, at his/her arrival to the facility, will find a line of at most *b* users, with a probability of at least α, where both *b* and α could be possibly different for each level and for each server. This is the quality of service condition.

Since we need allocation variables, instead of formulating the model using the Location Set Covering Problem equations, we use the traditional set of constraints for the plant location problem, rewritten for a two-level hierarchy. These constraints are a mandatory allocation constraint, and constraints forcing a demand to be served only at those places where there are servers, for both levels. Additionally, and also for both levels of service, we add constraints for the quality of service. With these constraints, the model is the following:

$$\text{Min } Z = \sum_j C_j w_j + \sum_k K_k z_k \qquad (1)$$

$$\sum_{j,k} x_{ijk} = 1 \qquad \forall i, \text{with } j \in N_i^l, k \in N_i^h, k \in M_j, \qquad (2)$$

$$x_{ijk} \le w_j \qquad \forall i, j,k \qquad (3)$$

$$x_{ijk} \le z_k \qquad \forall i, j,k \qquad (4)$$

P[low-level server j has $\le b$ people on queue] $\ge \alpha \quad \forall j \quad (5)$

P[high-level server k has $\le b$ people on queue] $\ge \alpha \quad \forall k \quad (6)$

$$x_{ijk}, w_{j,} \, z_k = 0, 1 \qquad \forall \, i,j,k \qquad (7)$$

where:

x_{ijk} = allocation variable that takes value 1 if population at demand node i is allocated to a low-level server located at the low-level candidate node j, and to a high-level server located at the high-level candidate node k, and zero otherwise.

w_j = location variable which takes value 1 if a low-level server is located at node j, and zero otherwise,

z_k = location variable that takes value 1 if a high-level server is located at node k, and zero otherwise,

C_j = cost of opening and operating a low-level service center at node j,

K_k = cost of opening and operating a high-level service center at node k,

$N_i^l = \{ j \mid d_{ij} \le S_{dl} \}$, the set of low-level candidate nodes located within S_{dl} of node i,

$N_i^h = \{ k \mid d_{ik} \le S_{dh} \}$, set of high-level candidate nodes located within S_{dh} of node i,

$M_j = \{ k \mid d_{jk} \le S_{lh} \}$, set of high-level candidate nodes located within S_{lh} of low-level candidate node j.

d_{ij} = shortest network distance between nodes i and j,

S_{dl} = standard distance from demand to low-level server,

S_{dh} = standard distance from demand to high-level server,

S_{lh} = standard distance from low-level server to its high-level server,

b = length of queue that is not to be exceeded with a predefined probability,

α = predefined probability of not exceeding the queue length b,

The objective (1) minimizes the cost of opening and operating the centers. Constraint (2) enforces mandatory allocation of each demand node to both low and high-level centers. Note that, in order to fulfill distance requirements and reduce the number of variables and constraints, variable x_{ijk} needs only be defined for $j \in N_i^l, k \in N_i^h, k \in M_j$. From now on, we assume that this is the case, so we do not need to repeat that the subscripts belong to these sets in the remainder of the paper. Constraints (3) and (4) assure that a demand node can not be allocated to a low or to a high-level candidate node unless there is a server located at it. Constraints (5) and (6) state that the queue length must be at most *b*, with probability α.

If the system is non-nested, the distance requirements establish linkages between the locations of the servers of different levels. Hence, locations of servers of both levels must be found jointly. In fact, in a non-nested system, if there were no distance restrictions relating the user (or demand) to the low-level servers and to the high-level servers, the problem could be cast as separated problems, one for each level. In the coherent case, there is the requirement that all the demands served by a particular low-level server must be served by the same high-level server. In a nested system, high-level servers provide also low-level services. These additional conditions, together with the distance requirements, make the problem non-separable.

3. Development of the Constraints for Quality of Service

The quality of service condition is enforced through constraints (5) and (6). These constraints must be written in an analytical form, preferentially as linear equations. This rewriting requires knowledge of the underlying probabilistic process. We model the system as a spatial queuing system, in which requests for service at each demand node appear according to a Poisson process. We also assume that service time distributes exponentially in servers at both levels.

Note that for both levels, each service center can have one or more servers. The equations describing queuing systems are different for the cases of one server and several servers, and so are the resulting constraints. We develop constraints for both situations, and, as an example, write a full model for the case in which the low-level centers have one server each and high-level centers have several servers. Models representing other configurations can be easily formulated using the same equations in a different order.

In this section, we use the following notation:

f_i = rate of appearance of requests for service at node *i*,

l_j^L = arrival rate of requests to low-level server *j*,
m_j^L = service rate at low-level server *j*,
$r_j^L = l_j^L / m_j^L$
l_k^H = arrival rate of requests to high-level server *k*,
m_k^H = service rate at high-level server *k*,
$r_k^H = l_k^H / m_k^H$
b_j = percentage of requests to low-level node *j* that request high-level service
p_s = probability of the queuing system being in state *s* (*s* users in the system),

For the low-level M/M/1 system, requests for service at each demand node *i* appear according to a Poisson process with intensity f_i. Since each low-level server is assigned to several demand nodes, the requests for service arriving at that low-level server are the union of all the requests for service of the demand nodes in its assignment set. Hence, they can be described as a second stochastic process, equal to the sum of several Poisson processes. This stochastic process can be easily shown to be also a Poisson process, with intensity l^L_j equal to the sum of the intensities of the processes at the nodes served by that server. This set of nodes is not known before the solution of the mathematical programming problem is obtained. However, we can use variables x_{ijk} in order to rewrite the parameter l^L_j as

$$l_j^L = \sum_{i,k} f_i x_{ijk} \qquad (8)$$

Using this definition, if a particular variable x_{ijk} is one, meaning that node *i* is allocated to server *j*, the corresponding intensity f_i will be included in the computation of l^L_j.

At server *j*, the exponentially distributed service time has an average service rate of m^L_j (with $m^L_j \geq l^L_j$, otherwise the system does not reach an equilibrium). If we assume steady state, we can use the well-known results for a M/M/1 queuing system for each low-level server and its allocated population.

We define the state *s* of the system as *s* users in the system (either being attended or in queue). That is, state zero corresponds to the server being idle and state *k* to one user being attended by the server and *k* - 1 in queue. We want the probability of a user being on a line with no more than *b* other people, being at least equal to α. This requirement is equivalent to saying that the probability of no people at the server's place, plus the probability of one person there, plus the probability of two persons, and so on, up to *b*+1 people at the server's place, must be at most α. If we represent as p_S the steady state probability of being in state *s*, this is written as:

$$p_0 + p_1 + \ldots + p_{b+1} \ge a \qquad (9)$$

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]:

$$p_s = (1 - r_j^L)(r_j^L)^s \ ,$$

where $r_j^L = l_j^L / m_j^L$. After replacing the values of p_S in equation (9) and some algebraic manipulation, we get

$$r_j^L \le \sqrt[b+2]{1-a} \, .$$

Since $r_j^L = l_j^L / m_j^L$,

$$l_j^L \le m_j^L \sqrt[b+2]{1-a} \, . \qquad (10)$$

Equation (10) is equivalent to constraint (5). Using equation (8), constraint (5) is rewritten as

$$\sum_{i,k} f_i x_{ijk} \le m_j^L \sqrt[b+2]{1-a} \qquad \forall j, \qquad (11)$$

which is a linear, deterministic equivalent of constraint (5), that forces the probability of people waiting in lines longer than *b*, to be of at most (1 - α). In this constraint, α could be different for each server.

In order to find the constraints for high-level servers, we find the intensities of the input processes, and show that these intensities follow the same kind of distribution as the input processes to the low-level servers.

The queuing diagram of the hierarchical system is the following:

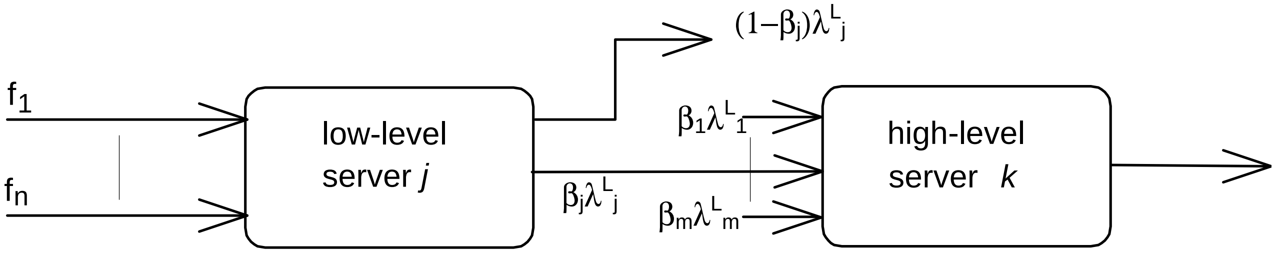

Figure 1: Queuing diagram of the hierarchical system

In the figure, f_1 to f_n are the intensities of the requests for service coming from nodes served by the low-level server *j*, and their sum is λ^L_j. The service rate of this server is μ^L_j. We assume that, at each low-level server *j*, a percentage β_j of the requests are referred to high-level servers. This percentage is either known or measurable. Thus, intensities $\beta_1\lambda^L_1$ to $\beta_m\lambda^L_m$ are the rates at which particular low - level servers pass requests to the high - level server, and their sum is λ^H_k. The high - level server serves them at a rate of μ^H_k.

In order to determine the input intensity of the high-level server, we first recall the equivalence property for M/M/1 and M/M/m queuing systems (Larson, Odoni, 1981). By this property, if the system has an infinite (or large enough) queue capacity, and an input arrival process of intensity λ^L_j, under steady-state conditions (that is $\lambda^L_j < m\ \mu^L_j$) the departure process is also Poisson, with the same intensity λ^L_j. Furthermore, the input process to high-level servers is the sum of several departure processes for low-level servers, with only some of the events being counted, where this selection is made at random (only requests that are referred to the higher level are counted). By virtue of these conditions, we can conclude that the input process to the high - level server *k*, λ^H_k, is also a Poisson process. The service rate of high-level servers is m^H_k.

Without loss of generality, and as opposed to the low - level server case, where we assumed a M/M/1 queuing system, for the high - level server we will assume a M/M/*m* server system. Thus, instead of the system having individual high - level servers, it has service centers, each with several servers. We do it so, in order to show how the probabilistic constraints are developed for such systems. As it will become clear, it is possible to use any combination of one - server / multiple - server systems for low - level / high - level facilities.

We use the results for a M/M/*m* queuing system for each center and its allocated users. Again, we define the state *n* of the system as *n* users in the system (either being attended or in queue). In the *m* servers case, if the system is in state *j*, with *j* ≤ *m*, all *j* users in the system are getting attention. In state *m* + *s*, however, *m* users are being attended and *s* in queue. We want to make the probability of a user finding a line with no more than *b* other people, at least equal to α. If we represent as p_n the steady state probability of being in state *n*, this requirement is written as:

$${}_0 \quad p_1 + ... + p_{m \;\; b} \quad a$$

b users at the arrival of the next

α. Conversely, since $_0$ p +.... + ¥ = 1,

$$p_{m+b+1} + p_{m+b+2} + ... + p_\infty \leq 1 - a \qquad (12)$$

which means that the probability of the queue being longer than *b* is smaller than (1 - α). 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 free server availability is desired, that is, at least one server free with probability α, then expression p_0 + p_1 +.... + p_{m-1} must be forced to be greater or equal to α.

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

$$p_n = p_0 (r^H)^n / \mathrm{n!} \qquad n \leq m$$

$$p_n = p_0 (r^H)^n \,/ (m!\, m^{n-m}) \quad n > m$$

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

Where ρ^H = l^H/m^H. Although these parameters are specific to each server center k, and for high-level servers, for the sake of simplicity we will not use any subscript (denoting node) or superscript (denoting server level) for the time being. With these expressions for the steady state probabilities, equation (12) becomes:

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

or

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

Since $\rho/m \le 1$, the summations in parentheses converge. Recalling that these summations can be written in a well known, simpler form, we get

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

After some algebraic manipulation, this equation becomes

$$\sum_{n=0}^{m-1}\frac{(m-n)m!m^b}{n!}\frac{1}{r^{m+b+1-n}} \ge \frac{1}{1-a} \qquad (13)$$

Since $\rho = \lambda/\mu$, and since λ is a function of the variables x_{ijk}, equation (13) can be also written as a function of variables x_{ijk}, becoming the deterministic equivalent of equation (6).

It is intuitively easy to see that, for any fixed value of α, the value of the left hand side of equation (13) can be made large enough to make the equation hold, by making ρ small enough, because its exponent is always positive. Manipulating variables x_{ijk}, (making as many of them equal to zero as needed) decreases the value of variable ρ. Furthermore, for any value of α there must exist a value ρ_a of ρ which makes equation (13) hold as equality, as well as a range of values of ρ such that equation (13) holds as a strict inequality.

Although it is the deterministic equivalent of equation (6), equation (13) can not be used in a linear model, because of its non-linearity. However, it is easy to see (and easy to show, by differentiating the left-hand side with respect to ρ) that its left-hand side (LHS) is strictly decreasing with increasing ρ, so we use this characteristic to find a linear equivalent to it.

Since the left-hand side is decreasing, there must exist a value ρ_a of ρ that makes equation (13) hold as equality for a given value of α. Since the LHS strictly increases when ρ decreases, for any value of $\rho \le \rho_a$, equation (13) also holds. Once the value of α is given, the value of ρ_a can be found by using any numeric root - finding technique (Newton methods, for example) on equation (13), written as an equality, and equation

$$r \le r_a \qquad (14)$$

becomes the new deterministic, linear equivalent of equation (13). This procedure is repeated for each service center *k,* and a value ρ_{ak} found for each one, obtaining the set of equations (where we now use also superscripts to mean that the equation applies to high-level servers)

$$r_k^H \le r_{ak}^H \qquad \forall k$$

Since $\rho^{H}_{k} = l^{H}_{k}/m^{H}_{k}$,

$$l_k^H \le m_k^H r_{ak}^H \qquad \forall k$$

Since λ^{H}_{k} is the intensity of a sum of several, selectively counted processes, it may be rewritten as a function of the intensities of these processes:

$$l_k^H = \sum_{j=1}^{M} b_j l_j^L ,$$

where the sum is over all the *M* low - level servers assigned to the high - level server *k*. As before, this sum can be written as

$$l_k^H = \sum_{i,j} b_j f_i x_{ijk} . \qquad (15)$$

Finally, the constraint is written as

$$\sum_{i,j} b_j f_i x_{ijk} \le m_k^H \; r_{ak}^H \qquad \forall k. \qquad (16)$$

The complete model with the constraints for quality of service is the following:

$$\text{Min } Z = \sum_{j} C_j w_j + \sum_{k} K_k z_k \qquad (1)$$

$$\sum_{j,k} x_{ijk} = 1 \quad \forall i, j, k \qquad (2)$$

$$x_{ijk} \leq w_j \quad \forall i, j, k \qquad (3)$$

$$x_{ijk} \leq z_k \quad \forall i, j, k \qquad (4)$$

$$\sum_{i,k} f_i x_{ijk} \leq m_j^L \sqrt[b+2]{1-a}\ , \quad \forall j \qquad (11)$$

$$\sum_{i,j} b_j f_i x_{ijk} \leq m_k^H \ r_{ak}^H \quad \forall\ k, \qquad (16)$$

$$x_{ijk}, w_{j,} \ z_k = 0, 1 \quad \forall\ i,j,k \qquad (7)$$

4. HiQ-MCLP: Hierarchical Queuing Maximum Covering Location Models

For the case in which serving all the population is not mandatory, we formulate the HiQ-MCLP model, that can be stated as follows:

"Maximize population covered by a two-level service, where a customer is considered as covered if she/he obtains low-level and high-level service within a pre-specified distance of her/his origin, not having to wait in a line with more than *b* other customers"

The difference between HiQ-LSCP and HiQ-MCLP is that in the later there is no mandatory coverage of all demands, but maximization of demand coverage is sought, when a predetermined number of servers is sited. The models for HiQ-LSCP and HiQ-MCLP differ in three constraints and the objective. In the first place, the mandatory allocation constraint (2) is substituted by a constraint that states that a demand node can not be allocated to more than one low-level and one high-level server. Secondly, the HiQ-MCLP includes constraints on the number of centers of both levels that can be located. Finally, the objective maximizes the coverage.

The Hierarchical Queuing Maximum Covering Location Model is the following:

$$\text{Max } Z = \sum_i \sum_j \sum_k a_i x_{ijk} \tag{17}$$

$$\sum_{j,k} x_{ijk} \leq 1 \qquad \forall i, j, k \tag{18}$$

$$x_{ijk} \leq w_j \qquad \forall i, j, k \tag{3}$$

$$x_{ijk} \leq z_k \qquad \forall i, j, k \tag{4}$$

$$\sum_{i,k} f_i x_{ijk} \leq m_j^L \sqrt[b+2]{1-a} \ , \qquad \forall j \tag{11}$$

$$\sum_{i,j} b_j f_i x_{ijk} \leq m_k^H \ r_{ak}^H \qquad \forall k, \tag{16}$$

$$\sum_j w_j = P_l \tag{19}$$

$$\sum_k z_k = P_h \tag{20}$$

$$x_{ijk}, w_j, z_k = 0, 1 \qquad \forall i,j,k \tag{7}$$

where the new parameters are:

a_i = the population at demand node *i*.,

P_l = the number of low-level centers to be located

P_h = the number of high-level servers to be located

In this model, the objective (17) maximizes the population (or customers) at node *i* receiving both low and high-level service. The new constraints are (18), (19) and (20). As stated above, constraint (18), which forces each demand to be assigned to at most one server at each level, replaces the mandatory coverage constraint (2), while constraints (19) and (20) set the number of centers of each level to be sited.

Constraints for Nested Systems

The formulations presented in preceding sections can be applied directly to nested systems, when modeled as co-located high-level and low-level servers. However, it is convenient to formulate the models in a form that makes explicit the low-level service provided by the high-level servers. This is achieved by modifying constraint (3) respectively to

$$x_{ijk} \leq w_j + z_j \qquad \forall i,j,k \tag{3'}$$

Furthermore, in nested systems there is often an additional requirement: customers attending a high-level site must receive both types of services there. An additional constraint must be added to reflect this requirement. This additional constraint is:

$$x_{ijj} = z_j \qquad \forall i, j \qquad (21)$$

Note that, in this case, high-level centers count also as low-level centers.

5. Extensions

Separate coverage for each level

Note that models could be written for separate coverage by both levels. This is done by replacing variables x_{ijk} with new variables x_{ij}, relating demand and low-level servers, and v_{ik}, relating demand and high-level servers. In this case, separate objectives could be written for coverage at each level. If the system is nested and If coverage is defined separately for each level, there is no need to force coverage of both levels at the same server. The model is the following:

$$\text{Max } Z = \quad w_L \sum_i \sum_j a_i x_{ij} + w_H \sum_j \sum_k a_i v_{ik} \qquad (22)$$

$$\sum_j x_{ij} \le 1 \qquad \forall\, i \qquad (23)$$

$$\sum_k v_{ik} \le 1 \qquad \forall\, i \qquad (24)$$

$$x_{ij} \le w_j + z_j \qquad \forall\, i,j \qquad (25)$$

$$v_{ik} \le z_k \qquad \forall\, i,k \qquad (26)$$

$$\sum_i f_i x_{ij} \le m_j^L \sqrt[b+2]{1-a} \;, \qquad \forall j \qquad (27)$$

$$\sum_i g_i f_i v_{ik} \le m_k^H r_{ak} \qquad \forall k \qquad (28)$$

$$\sum_j w_j = P_l \qquad (29)$$

$$\sum_k z_k = P_h \qquad (30)$$

$$x_{ij},\, v_{ik},\, w_j,\, z_k = 0, 1 \qquad \forall\, i,j,k \qquad (31)$$

Note that we need to define a new parameter γ_i, the percentage of the customers from node i that request high-level service. A reasonable estimator for this parameter is b_j, when node i is allocated to low-level server j. Note that the problem is still non-separable because of constraint (25). Parameters w_l and w_h are weights on each objective. Therefore, the formulation is cast as a multiobjective problem.

Coherent Systems

So far it has been assumed that the hierarchical organization is not coherent, as defined in Serra and ReVelle (1991). In a coherent hierarchical system, all demands allocated to a particular low-lever server are also assigned to the same high-level server (*coherent*). In order to enforce the coherence requirement, we define a new variable y_{jk}, as being equal to 1 if demands assigned to low - level server j, are also assigned to high - level server k. In other words, it relates a low - level server to a high - level server. Also, for nested systems, we need the new constraint:

$$y_{jj} = z_j \qquad \forall j \qquad (32)$$

forcing customers to receive both levels of service at the high-level server. The new formulation is the following:

$$\text{Min } Z = \sum_j C_j w_j + \sum_k K_k z_k \qquad (1)$$

$$\sum_{j,k} x_{ijk} = 1 \qquad \forall\, i \qquad (2)$$

$$x_{ijk} \le y_{jk} \qquad \forall\, i,j,k \qquad (33)$$

$$y_{jk} \le z_k \qquad \forall\, j,k \qquad (34)$$

$$y_{jk} \le w_j \qquad \forall\, j,k \qquad (35)$$

$$\sum_k y_{jk} = 1 \qquad \forall\, j \qquad (36)$$

$$\sum_{i,k} f_i x_{ijk} \le m_j^L \sqrt[b+2]{1-a}\ , \qquad \forall j \qquad (11)$$

$$\sum_{i,j} b_j f_i x_{ijk} \le m_k^H\ r_{ak}^H \qquad \forall\, k, \qquad (16)$$

$$x_{ijk},\ w_j,\ z_k,\ y_{jk} = 0, 1 \qquad \forall\, i,j,k \qquad (37)$$

Objective (1) and constraint (2) are the same as before. Constraint (33) states that it is not possible for a demand to be assigned both to low - level server *j* and high - level server *k*, unless they are related by a variable y_j. The meaning of constraints (34) and (35) is that candidate nodes of both levels can not be related unless there are servers at both of them. Constraint (36) forces all low - level servers to relate to one and only one high - level server. By virtue of this constraint, together with constraint (33), all demands assigned to a low - level server are assigned to the same high - level server. All remaining constraints are the same as before.

In order to reduce the number of constraints, constraint (33) might be replaced by

$$\sum_i x_{ijk} \le |I| y_{jk} \qquad \forall\, j,\, k. \qquad (38)$$

where I is the total number of demand nodes. However, this constraint is less tight, and if solved as a relaxed linear programming problem, it could lead to the occurrence of more fractional integer-defined variables in the solution.

For coherent systems, the maximal covering model is:

$$\text{Max } Z = \sum_i \sum_j \sum_k a_i x_{ijk} \qquad (17)$$

$$\sum_{j,k} x_{ijk} \le 1 \qquad \forall\, i \qquad (18)$$

$$x_{ijk} \le y_{jk} \qquad \forall\, i,j,k \qquad (33)$$

$$y_{jk} \le z_k \qquad \forall\, j,k \qquad (34)$$

$$y_{jk} \le w_j \qquad \forall\, j,k \qquad (35)$$

$$y_{jj} = z_j \qquad \forall\, j \qquad (32)$$

$$\sum_k y_{jk} \le 1 \qquad \forall\, j \qquad (39)$$

$$\sum_{i,k} f_i x_{ijk} \le m_j^L \sqrt[b+2]{1-a}\ , \qquad \forall j \qquad (11)$$

$$\sum_{i,j} b_j f_i x_{ijk} \le m_k^H\ r_{ak}^H \qquad \forall\, k, \qquad (16)$$

$$\sum_j w_j = P_l \qquad (19)$$

$$\sum_k z_k = P_h \qquad (20)$$

$$x_{ijk},\, w_j,\, z_k,\, y_{jk} = 0,\, 1 \qquad \forall\, i,j,k \qquad (37)$$

Minimum distances

Finally, a secondary objective could be added to these models, which minimizes the distances from the demand to both levels of servers, and the distances between the two levels of servers for each demand. This objective has the form:

$$\min(\sum_{i,j,k} a_i d_{ij} x_{ijk} + \sum_{i,j,k} a_i d_{ik} x_{ijk} + \sum_{i,j,k} a_i d_{jk} x_{ijk}),$$

where a_i is the population at demand node *i*.

6. A Heuristic Procedure to Solve the Models

The models presented in the previous section have the common characteristic of having thousands of variables and constraints for relatively small networks. Therefore, the use of traditional optimal solution methods such as linear programming plus branch and bound can become very burdensome in terms of computing times and for relatively large networks these methods cannot be applied. On the other hand, the deterministic constraints create an additional problem in finding integer solutions since the specified parameters are not equal to 0 or 1. This implies that the number of branches is likely to increase dramatically (see ReVelle, 1993, on Integer Friendly Programming).

Therefore, it is necessary to develop some alternative solution procedures to solve these problems. In this section we offer a bi-level heuristic for the Nested Hierarchical Queuing Maximal Covering Location Model (HiQ-MCLP) that has two phases: a construction phase and an improvement phase.

In the first phase (construction phase), a greedy adding procedure with random substitution (GRASP) is used to find the facilities of the first hierarchical level, where in each iteration the vertex with the best objective value for the first level is added to the set of locations. Once these locations are found, complete enumeration is used to find the optimal locations of the second level. This procedure is feasible even for large networks, since the model is nested and therefore candidate nodes for the second level facilities are restricted to the actual first level facilities. In each instance of the enumeration procedure, the overall objective is computed and the best solution is stored.

In the second phase, the heuristic will try to find a better solution. For each one-opt exchange of facilities at the first level, complete enumeration is used again to find the locations at the second level and in each instance of the enumeration the objective is computed and the best solution for both levels is stored. Observe that for each exchange of the first level facilities, the locations of the second level are completely recalculated. In other words, there is no "memory" concerning the locations of second level facilities.

If the relocation of one first level facility and the new locations of the second level facilities has provided a set of positions that is better than before the one-opt trade, it will keep the new set of locations as the best so far. Otherwise, the procedure will ignore the relocation and will restore the previous solution. The one-opt trade will be done for all nodes and first level facilities and repeated until no cycle results in an improvement.

Since the VS phase only considers vertices that improve the objective, the heuristic may end in a local optimum. In order to avoid being trapped in a local optimum, a tabu search procedure is developed, similar to the one presented by Benati and Laporte (1994) (TABU phase). In essence, this tabu search explores a part of the solution space by repeatedly examining all neighbors of the current solution, and moving to the best neighbor even if this causes the objective function to deteriorate. To avoid cycling, recently examined solutions are inserted in a constantly updated tabu list. At each iteration, a first level facility is selected, the *m* vertices that are closest to it are considered candidate nodes for it. For each of the candidates the objective is computed (using the same procedure as in the VS procedure, that is, finding the best second-level facilities with complete enumeration) and the one that is not declared tabu with the highest objective is chosen. If the value of the new solution improves the objective, the new solution is stored as the best one, and the vertex where the facility has moved to is declared tabu for *t* iterations. Otherwise, the new solution is still implemented but it is not considered as the best solution so far. If all neighbor vertices are declared tabu, then the one with the lowest tabu tag is chosen as the new solution. The number of one-opt trades needs to be fixed *a priori*.

Once the number of one-opt trades is reached, the tabu procedure is restarted using as initial solution the p_l nodes that were least visited in the previous tabu phase. This is now as the diversification step.

As mentioned before, this heuristic is useful when locating nested hierarchical facilities using a maximal covering approach. If services are non-nested, the use of complete enumeration to find the locations for the second level facilities can be very burdensome in terms of computing time, since number of candidate nodes becomes much larger. In this case, we can also use the vertex substitution heuristic in the second level together with the tabu phase.

If we are solving a Hierarchical Queuing Location Set Covering Problem (HiQ-LSCP), we do not know how many facilities we are going to locate. Therefore, the heuristic has to be modified in the following way: Solve a standard Location Set Covering Problem (LSCP) in each level separately. This will set a lower bound on the number of facilities needed to cover the population at the first level and at the second level, p_l^* and p_h^* respectively. Then the heuristic can be used to find a solution with this values of p_l^* and p_h. If a feasible solution is not found for the first level, meaning that it is not possible to serve all the demand with adequate quality of service, p_l^* is increased in one unit ($p_l^* = p_l^* +1$) and the model is solved again. In a similar matter, if no feasible solution for the second level is found, the value of p_h^* is increased in one unit. This procedure is applied until a feasible solution is found for both levels.

7. Computational Experience

In this section the heuristic described above is used to solve the nested HiQ-MCLP. First, in order to test the performance of the heuristic, 1000 randomly generated networks were used with different values in some of the parameters. The networks were generated following the method described by Cordeau et al (1997) to obtain networks with the anisotropic characteristics that normally exist in real geographical spaces. Here, f is a constant equal to 0.05, and n and t are given as input data (n is the number of nodes and t represents a small number between 4 and 8, depending on the number of nodes):

1. Randomly generate t centers in the $[-50,50]^2$ square according to a continuous uniform distribution.
2. Set $i := 1$.
3. While *i* £ *n*, do:

- Randomly generate a vertex v_i in the $[-100,100]^2$ square according to a continuous uniform distribution and compute its nearest distance d to the nearest center.
- Let u be a number randomly chosen in the [0,1] interval according to a continuous uniform distribution. If *u* £ e^{-fd} , set $i := i + 1$. Otherwise, delete v_i.

Each vertex is as the same time a demand node and a potential facility site. First level facilities have only one server, while second level facilities have m servers As for the demand at each node, this one was computed using a random uniform distribution within the range [90,110]. For each level the coverage distance S_d was computed as follows:

$$S_d = DISTMAX/(2*np)$$

where *DISTMAX* is the maximum distance between any 2 nodes of the network, and *np* is the number of facilities to locate.

For the Tabu phase, the number of iterations was arbitrarily set to the number of nodes multiplied by the number of first level facilities to locate. The tabu tag was set to a random value within the [4,8] interval. For each potential facility, the neighbors were chosen as the s closest vertices, s randomly chosen between 4 and 8.

For each run, both the heuristic presented and complete enumeration were used to obtain solutions and test the heuristic.

Results are presented in Table 1. The first column, labeled as **ND**, corresponds to the number of nodes in the network. The number of facilities to locate in each level is presented in the second column (**np1, np2**). Then, the third column sets the value of α. The rest of the parameters was set as follows**:**

Parameter	Value
Intensity factor	0,00162
Max number of people waiting in the first level	2
Max number of people waiting in the second level	2
Service rate at the first level	4
Service rate at the second level	2
Number of centers in the second level	2
Proportion of referral	45%

For each ND, np1, np2 and α 100 networks were generated. In the fourth, fifth and sixth columns the number of optimal solution that were found in each of the phases of the algorithm is presented. The next column shows the total number of non-optimal solutions. Next, the average and maximum deviations from optimality are presented. Finally, computer times for both the complete enumeration procedure and the heuristic are shown.

The heuristic achieved relatively good results in obtaining optimal solutions, since only 6% of the total runs were non-optimal. The largest deviation from optimality was equal to 9.9% and the average deviation ranged between 0.3% and 3.3%.

An example

The heuristic was also used in the well-known Swain's 55-node test network (Swain, 1974, see Appendix). The parameters used were set as follows:

Parameter	Value
Number of centers in the first level	4
Number of centers in the second level	2
Intensity factor	0,00166
Service rate at the first level	4
Service rate at the second level	2
Proportion of referral	45%

Results are presented in Table 2. In the first column the maximum number of customers waiting in line for each level is presented. The second column sets the α reliability level. The final population covered is shown in the third column.

Then, for each level, the final locations are presented together with the total frequency in parenthesis (left hand side of the deterministic constraint) and the right hand side of this equation. As expected, as the desired maximum number of customers waiting in line is reduced, and as the α reliability level is increased, the total population covered decreases, since the deterministic constraints become tighter.

References

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

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

Batta R., Dolan J., and Krishnamurthy N., 1989: “The maximal expected covering location problem: Revisited”, *Transportation Science*, **23**, 277-287.

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

Benati S., Laporte G., 1994: “Tabu search algorithms or the (r/Xp)-medianoid and (r/p)-centroid problems”, *Location Science*, **2**, 193-204.

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

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.

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

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

Church R L., Eaton D J., 1987: "Hierarchical Location Analysis Using Covering Objectives", in *Spatial Analysis and Location-Allocation Models* (A. Ghosh and G. Rushton, eds), Van Nostrand Reinhold, New York.

Cordeau, Gendreau, Laporte G., 1997: “” Networks, **30**, 105-119.

Daskin M., 1983: “A maximum expected covering location problem: Formulation, properties, and heuristic solution”, *Transportation Science*, **17**, 48-70.

Gerrard R.A. Church R.L., 1994: “A Generalized Approach to Modeling the Hierarchical Maximal Covering Location Problem with Referral, Papers in *Regional Science*, **73(4)**; 425 – 454.

Hogan K., ReVelle C., 1986: “Concepts and applications of backup coverage”, *Management Science*, **32**, 1432-1444.

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

Mandell M., 1966: "Covering Models for Two-tiered Emergency Medical Services Systems", ISOLDE VII, Edmonton/Jasper, Canada, June 26 - July 3, 1996.

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

Marianov V., ReVelle C., 1996: "The Queuing Maximal Availability Location Problem: a Model for the Siting of Emergency Vehicles", *European Journal of Operations Research*, **93**, 110 - 120.

Marianov V., Serra D., 1998: "Probabilistic Maximal Covering Location-Allocation Models for Congested Systems", *Journal of Regional Science*, **38** (3), 401 - 424.

Narula S C., 1985: "Hierarchical Location-Allocation Problems: A Classification Scheme", *European Journal of Operations Research*, **15**, 93 - 99.

Parzen E., 1962: *Stochastic Processes*, Holden-Day Series in Probability and Statistics, Holden-Day.

ReVelle C., 1993: “Facility Siting and Integer Friendly Programming", *European Journal of Operations Research*, **65**, 147 - 158.

Serra D., ReVelle C., 1993: “The pq-median problem: Location and districting of hierarchical facilities”, *Location Science*, **1**, 299-312.

Serra D., ReVelle C., 1994: “The pq-median problem: Location and districting of hierarchical facilities-II. Heuristic solution methods”, *Location Science*, **2**, 63-82.

Serra D., 1996: "The Coherent Covering Location Problem", *Papers in Regional Science: The Journal of RSAI*, **75**, 1: 79 - 101.

Serra D., Marianov V., ReVelle C., 1992: "The Maximum Capture Hierarchical Problem", *European Journal of Operations Research*, **62**, Nº 3, 363 - 371.

Swain R, 1974: “A parametric decomposition algorithm for the solution of uncapacitated location problems”, *Management Science* **21**, 189-198.

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

Table 1: Computational Results.

	np1,		# of opt solutions found			# of	Avg.	Max.	Comp. Time	
ND	**np2**	**α**	**Grasp**	**T&B**	**Tabu**	**non opt**	**Dev**	**Dev**	**Enum**	**Heur**
30	3,2	85%	20	45	34	1	0.3%	0.3 %	0.006	0.011
		95%	4	44	52	0	0.0%	0.0%	0.006	0.012
	4,2	85%	10	70	19	0	2.1%	2,1%	0.033	0.039
		95%	20	50	30	0	0%	0%	0.035	0.038
40	4,2	85%	12	38	42	8	2.6%	9.9%	1.13	0.19
		95%	11	42	37	10	3.0%	7.8%	1.16	0.18
	5,3	85%	23	33	33	11	1.2%	4.9%	16.01	0.57
		95%	17	31	43	9	3.3%	9.8%	17.02	0.58
50	4,2	85%	15	38	38	9	2.2%	8.1%	3.40	0.27
		95%	14	38	36	12	2.4%	9.8%	3.56	0.25

Table 2: Example on a 55-node network

		Pop	**Level 1**		**Level 2**	
NB1,NB2	α	**Cov**	**Locations (cap)**	**RHS**	**Locations**	**RHS**
3,3	**85%**	2750	**8**(2.20) **10**(0.93) **17**(0.81) **36**(0,84)	2,73	**10**(1.47) **17**(0.90)	2.84
3,3	**90%**	2744	**4**(2.09) **19**(0.94) **36**(0.91) **38**(0.62)	2.52	**36**(1.37) **38**(1.01)	2.64
3,3	**95%**	2703	**1**(1.85) **18**(0.94) **33**(0.82) **41**(0.82)	2.20	**33**(1.00) **41**(1.48)	2.33
2,2	**95%**	2671	**29**(1.15) **32**(0.68) **41**(1.12) **44**(1.42)	1.89	**32**(0.97) **41**(1.53)	2.08
3,3	**99%**	2504	**18**(0.94) **41**(1.04) **44**(1.20) **45**(0.92)	1.59	**41**(1.21) **45**(1.19)	1.76
1,1	**95%**	1768	**11**(0.98) **13**(0.93) **25**(0.64) **55**(0.39)	1.47	**25**(1.38) **55**(0.86)	1.74