RAIRO Operations Research

RAIRO Oper. Res. **39** (2005) 143–162

DOI: 10.1051/ro:2006002

ALLOCATING SERVERS TO FACILITIES, WHEN DEMAND IS ELASTIC TO TRAVEL AND WAITING TIMES *

Vladimir Marianov[1], Miguel Rios[1] and Francisco Javier Barros[2]

Abstract. Public inoculation centers are examples of facilities providing service to customers whose demand is elastic to travel and waiting time. That is, people will not travel too far, or stay in line for too long to obtain the service. The goal, when planning such services, is to maximize the demand they attract, by locating centers and staffing them so as to reduce customers' travel time and time spent in queue. In the case of inoculation centers, the goal is to maximize the people that travel to the centers and stay in line until inoculated. We propose a procedure for the allocation of multiple servers to centers, so that this goal is achieved. An integer programming model is formulated. Since demand is elastic, a supply-demand equilibrium equation must be explicitly included in the optimization model, which then becomes nonlinear. As there are no exact procedures to solve such problems, we propose a heuristic procedure, based on Heuristic Concentration, which finds a good solution to this problem. Numerical examples are presented.

Keywords. Facility location, resource allocation, nonlinear optimization, integer programming, heuristics.

Mathematics Subject Classification. 90B80, 90B22.

Received December 10, 2004. Accepted September 19, 2005.

* *This research has been possible partly due to grant FONDECYT 1020778.*

[1] Department of Electrical Engineering, Pontificia Universidad Catolica de Chile, Vicuna Mackenna 4860, Santiago, Chile; marianov@ing.puc.cl; mrios@ing.puc.cl

[2] Graduate Program, Department of Systems Engineering, Pontificia Universidad Catolica de Chile, Vicuna Mackenna 4860, Santiago, Chile; FBarros@sqm.cl

1. Introduction

Many business activities, private or public, including product sales or service delivery, are performed at supplier facilities. In many cases, clients have to travel to these facilities and wait in line in order to be attended. Banks, clinics or inoculation centers, distribution centers, gas stations and distributed ticket selling facilities are all examples of this type of situation. In these cases, in addition to the cost of the goods or service, consumers must incur in travel and waiting costs.

The waiting time responds to a congestion externality. Externality exists when the utility of a consumer is affected positively or negatively by the decisions and actions of other consumers. Examples of negative congestion externalities include waiting time in service facilities, traffic delay in transportation systems, blocking probability in telecommunication systems (probability of not being able to complete a call because of congestion in the system), bed occupancy level in hospitals and noise pollution in residential areas. A positive congestion externality happens, for example, in pubs or dancing clubs, where patrons are crowd-lovers. When there are negative externalities and the service is essential, customers' only option is waiting in a queue until served. If the service is not essential, some arriving clients may choose not to wait if there is a line, rejecting the service. In this latter case, it is said that the demand is elastic because it is affected by the externality produced by congestion.

The demand may also be elastic to travel distance or travel time to the facility, or to other parameters that bear no relation to the location of the facilities or the congestion, such as price, quality of service, availability of parking space, attendants' qualification, quality of the decoration and furniture and, in general, all those features that make visiting the center attractive to clients. When the demand is elastic, providers attract more clients by offering lower prices, nearer facilities, shorter lines or a better service. If consumers can choose in what facility to get the goods or services, the system is known as a "user-choice" system. In "planner-choice" systems, it is the planner who decides the assignment of customers to facilities, which is the case of the systems in this paper.

In systems dealing with demand that is elastic to waiting time, the use of a facility depends on the aggregate decision of all the users, and the length of the queues at each facility, as well as the total demand served, depend on the equilibrium the system reaches (if there is such equilibrium). In fact, as the demand grows and more customers decide to stay in line at a certain facility, the lines become longer, and the demand on that facility decreases because of the elasticity. In turn, since the demand decreases, the line length decreases until equilibrium is reached.

If there is no queuing at the facilities, one server (clerk, attendant) is enough to serve all the clients arriving to a facility. However, if there is congestion, more than one server could be needed in order to serve customers within reasonable time limits. The allocation of an adequate number of attendants to facilities has influence on the waiting time of the clients and the length of the queue they form. In fact, if a center has more attendants (at a higher cost), the waiting time (or queue length) of the clients is reduced. Because of the elasticity, this will again increase the demand on the facility, until equilibrium is reached. If there is a budget limit, the problem becomes how to allocate a limited number of servers, p, to existing facilities.

In this paper, we propose a new model that allocates servers to existing facilities. In order to formulate this model, a multiple-server queuing system is considered, that is seldom used in literature because of its complexity. The number of servers in each facility is determined by the integer model. Since the demand is elastic, its total amount at each facility is also computed by the model. Therefore demand equilibrium equations are added, which make the problem nonlinear. We also propose an *ad-hoc* heuristic procedure for finding good server allocations. This model and procedure are applicable to systems in which facilities are subject to queuing effects, when the demand is elastic to travel and waiting time. The solution of this model also indicates which customers are assigned to which facility, so that the demand is maximized.

The remainder of the paper is organized as follows: Section 2 presents a literature review. Section 3 presents the development and formulation of the model; Section 4, the heuristic procedure; Section 5 the computational experience and Section 6, the conclusions.

2. Literature review

The problem of locating p facilities (or servers) has been studied exhaustively, at least when at most one facility (or server) is allowed at any candidate point. Hakimi [9] stated the p-median problem and proved that at least one optimal solution for this problem consists of locating the p facilities at nodes of the network. Later, ReVelle and Swain [20] introduced the integer linear programming formulation for the p-median, and studied its integer properties. The problem of allocating servers appears only when there is congestion, as in the case of airport landing stripes [18], or when the service needs to be provided by more than one server at a time, as in the case of fire-fighting equipment [13, 14].

Most real problems require considering stochastic demand [16]. In Berman and Krass [2] location problems are modeled in a general form as Location Problems with Stochastic Demands and Congestion (LPSDC). The authors identify two types of problems, those of the cover-type and those of the median-type. The former includes the Set Covering problems and the Maximal Weight Covering problems. The median-type problems have a formulation which consists of a highly nonlinear objective function with linear restrictions. For this last problem, it is only possible to find heuristic solutions.

Several examples of cover type LPSDC problems are found in the literature. For instance, Marianov and Serra [17] formulate a model that minimizes the number of facilities to locate and the amount of servers to assign to each installation (multi-objective formulation), given that the facilities can become congested. The model allows the designer to trade-off investment and operating costs against quality of the service. The model is solved using Heuristic Concentration (HC).

Marianov and Serra [18] address a Set Covering LPSDC. They optimally locate hubs on airline networks, and allocate landing strips to hubs. The heuristic procedure they use is based on Greedy Adding, One-Opt interchanges and Tabu Search.

Laporte *et al.* [12] propose the Capacity Location Problem (CLP). They consider the problem of optimally determining the location and the size of the facilities, when the demand of the clients is uncertain. In order to solve it, they use Benders decomposition and a Branch & Cut based algorithm.

In terms of the median-type LPSDC models, Berman and Mandowsky [6] consider the problem of simultaneously finding the districting policy and the optimal location of mobile servers. The heuristic is based on a two-server location heuristic and a districting procedure from Berman and Larson [3]. If the initial point is good (usually it starts from a p-median solution), this heuristic is very efficient. Berman *et al.* [5] improve on the results of Berman and Mandowsky [6] by relaxing the districting and allowing servers districts to overlap. The solution procedure uses Larson's Hypercube model and improves the location of each facility by solving a 1-median problem and a Stochastic Queue Median problem.

A capacitated median-type problem is presented in Zhou and Baoding [27]. Three types of model are proposed: an Expected Value Model, a Chance Constrained Programming model and a Dependent-Chance Programming model. To solve these problems, the authors propose a hybrid algorithm based on simplex, stochastic simulation and genetic algorithms.

In Marianov and Serra [19] a median-type model is proposed to locate a fixed number of facilities with multiple servers in a congestible network. The model has multiple objectives: minimize the travel cost and the expected congestion cost. It is solved using a Tabu Search and Ant-Colonies based meta-heuristic.

Berman and Krass [2] study four median-type LPSDC problems. The first model locates a single facility. The second model locates p facilities, each containing a mobile server with non-cooperative characteristics. This model is based on distributing exclusive service areas. In the third model the servers can cooperate between them, and the Hypercube model is used for solving the problem. The last model is median-type and it locates any number of stations, in which mobile servers respond to the calls from wherever they are located, *i.e.*, a server only returns to its station if there are no calls requiring service. This model is only solved for the first case presented in Berman and Vasudeva [4], that is, only a server and one station are located. The authors recognize that problems in which demand is elastic to congestion are very difficult.

A very good and exhaustive review of stochastic demand models is presented by Brandeau *et al.* [7]. This review covers the cases in which demand is elastic and inelastic to congestion and/or travel time, and covers also different demand – facility assignment schemes (planner choice or user choice). In all the proposed models the congestion is measured by the length of the line in an M/G/1 system (one server). In [7], a model is offered that locates p facilities in a congested network, with planner choice, so that the total utilization of the facilities is maximized. The problem can be reduced to a traditional p-median with an appropriate demand balance. However the model is not solved in the paper, and the authors outline a numerical approach for the case in which some convexity results can be proven.

A model for location of fixed facilities with multiple-servers on a network and for allocation of elastic demand is proposed by Marianov [15]. The model maximizes the total demand served, locates several facilities with multiple servers in each one, uses a practical demand curve, and measures congestion by counting the number of clients in the system, as opposed to using waiting time as proposed in Brandeau *et al.* [7]. A special heuristic solves this integer and nonlinear optimization problem, which is a combination of Lagrangean relaxation and an iterative scheme. However, this model considers a fixed number of servers in each center. In other words, it does not consider server allocation. Furthermore, the heuristic is not modifiable to solve the problem presented in this paper.

3. Model development and formulation

Let $G(N, A)$ be an undirected network with node set N and arc set A. Let I denote the set of demand nodes and J the set of feasible location sites. Each node of the network represents either a demand concentration or a feasible site to put a facility, or both. The maximum potential demand $\bar{f}_i$ at each demand node i is known. However, the actual demand originating at this node and attended at the assigned facility is smaller than $\bar{f}_i$, due to the elasticity to travel and waiting time.

The travel time from demand node i to facility at node j is assumed proportional to the distance between i and j, measured over any path (for example, the shortest path). We make the usual assumption that all customers choose the same path. We represent demand elasticity to the distance from i to j through a parameter $\beta_{ij} \in$ [0, 1] that can be obtained from focus groups studies, data polls, or behavioral studies. This parameter decreases with distance and it represents the drop in demand with increasing travel time from node i to node j. An example of a parameter β_{ij}, is shown in Figure 1. In this example, the parameter is linearly decreasing with distance, but it could have any decreasing shape.

In the example of Figure 1, d_{ij} is the travel distance (in distance units) from node i to a facility j. The points in the graph represent the values of the parameter for all possible distances between demand node i and facility locations j. Note that intermediate values are not needed. The only requirement on the parameter in order to apply our method is for it to be decreasing with distance. Consequently, the graph can be nonlinear, discontinuous or have any shape, as long as the values decrease with distance.

Similarly, the waiting time at a facility is proportional to the queue length at that facility. The queue length is in turn equal to the number of clients exceeding the number of servers at that facility. The demand elasticity to queue length can be represented using the parameter $\gamma_n \in$ [0, 1], also obtained from experiments or practical studies. This parameter represents how the maximum potential demand $\bar{f}_i$ will decrease when there are n persons waiting to be served. This parameter is the same for all demands and facilities, and it depends only on the queue length.

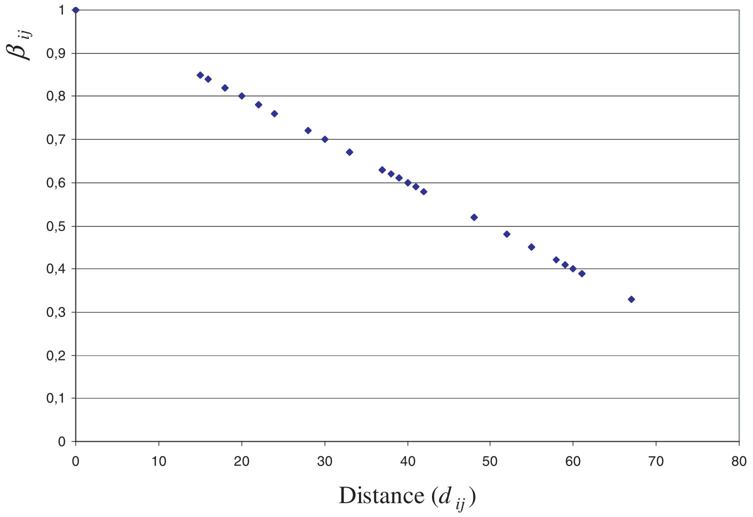

FIGURE 1. Example of parameter β_{ij} linearly decreasing with distance.

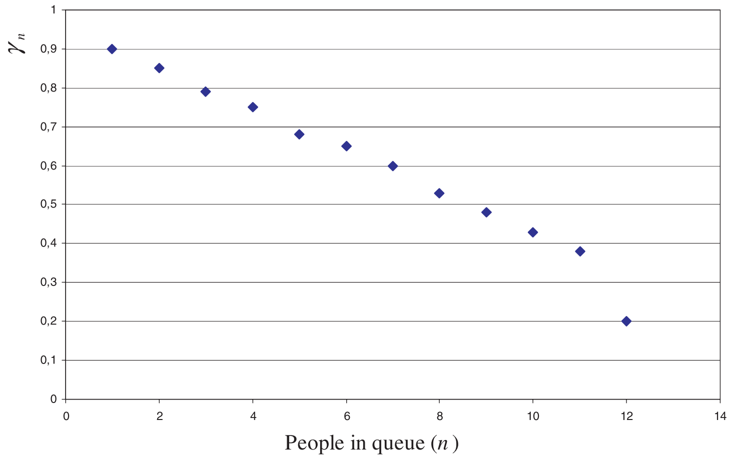

FIGURE 2. Example of parameter γ_n.

Theoretically, it could also depend on the demand node i, but this would make the problem more difficult to solve.

Figure 2 shows an example of parameter γ_n. In this case, the shape is not linear. Again, the only requirement on this parameter is for it to be decreasing with n.

We assume that requests for service at each node i appear according to an independent Poisson process with mean $\bar{f}_i$. All the demand originating at a node is assigned to the same facility by the planner, seeking to maximize the total demand served. This typically happens with inoculation centers, voting centers, police stations, and so on. Variable x_{ij} is equal to 1 if the demand at node i is assigned to the facility at j. Since the demand is elastic, the percentage of the total demand that goes from i to j and waits in line until served, assuming that on its arrival to the facility the queue length is n, is

$$\lambda_{ijn} = \beta_{ij} \cdot \gamma_n \cdot \bar{f}_i \cdot x_{ij}.$$

The expected value of the demand from i that is served at j, over all queue lengths is:

$$\lambda_{ij} = \sum_n \lambda_{ijn} P_{nj} = \sum_n \beta_{ij} \cdot \gamma_n \cdot \bar{f}_i \cdot x_{ij} \cdot P_{nj} \tag{1}$$

where P_{nj} is the probability of finding n clients at facility j. Finally, requests for service at each facility j are the sum of all requests originating at the demand nodes assigned to facility j. Recall that processes at demand nodes are Poisson distributed. At demand nodes, this distribution is a good approximation of what happens in reality. However, since clients can refuse the service, and the refusal is somehow related to the expected waiting time at the facilities, the process at the facilities might not be exactly exponentially distributed. However, better approximations may become intractable so, as a first approximation, and since no other approximations are available to this problem, we assume that the process at each facility is the sum of several independent Poisson processes and is itself a Poisson process of rate λ_j.

The probability P_{nj} depends on the mean arrival rate of clients, the mean service rate and the number of servers at the facility. For the case under consideration, the number of clients at each facility will be assumed to be limited to K, because of space limitations. Therefore, the total demand served by the center at j is:

$$\lambda_j = \sum_{i \in I} \lambda_{ij} = \sum_{i \in I} \sum_{n=0}^{K} \beta_{ij} \cdot \gamma_n \cdot \bar{f}_i \cdot x_{ij} \cdot P_{nj}. \tag{2}$$

We assume that consumers assign different values to travel and service times. Therefore, both are treated separately in our model. We also assume exponentially distributed service time, with a mean service rate μ_j for each server located at j. If we assume steady state, we can use the known results for an $M/M/s/K$ queuing system [10] for each center and its allocated users.

The probabilities P_{nj} of each state in this queuing system are:

$$P_{0j} = \left[1 + \sum_{n=1}^{s} \frac{\lambda_j^n}{\mu_j^n \cdot n!} + \frac{\lambda_j^s}{\mu_j^s \cdot s!} \sum_{n=s+1}^{K} \left(\frac{\lambda_j}{\mu_j \cdot s} \right)^{n-s} \right]^{-1} \tag{3}$$

$$P_{nj} = \begin{cases} \frac{\lambda_j^n}{\mu_j^n \cdot n!} \cdot P_0 & \text{for } n \leq s \\ \frac{\lambda_j^n}{\mu_j^n \cdot s! \cdot s^{n-s}} \cdot P_0 & \text{for } s \leq n \leq K \\ 0 & \text{for } n \geq K. \end{cases} \tag{4}$$

Note that P_{nj} depends on the mean arrival rate to that node, λ_j, the mean service rate at the node μ_j, the maximum capacity K of the node, and the number of servers allocated to center j, which is variable and we will call y_j. Both the mean service rate and the capacity are given parameters. However, λ_j and y_j are variables determined by the solution of the problem. Therefore equation (2) becomes:

$$\lambda_j = \sum_{i\in I} \lambda_{ij} = \sum_{i\in I}\sum_{n=0}^{K} \beta_{ij}\cdot\gamma_n\cdot\bar{f}_i\cdot x_{ij}\cdot P_{nj}(\lambda_j, y_j). \tag{5}$$

This equation represents the supply-demand equilibrium at each node j. As the mean arrival rate λ_j to the node increases, so does the queue length and waiting time at that center. In turn, a longer queue length at the facility makes the mean arrival rate to decrease because of the elasticity of the demand, causing a reduction of the queue length and waiting time, and an increase in the arrival rate λ_j. In order to prove that such equilibrium exists, note that equation (5), can also be written as:

$$\lambda_j = g(\lambda_j, x_{ij}, y_j) \qquad \forall j\in J$$

or

$$F(\lambda_j) = g(\lambda_j, x_{ij}, y_j) - \lambda_j = 0 \qquad \forall j\in J. \tag{6}$$

Lemma. *The function $F(\lambda_j)$ has a unique root.*

Proof. Recall that

$$g(\lambda_j, x_{ij}, y_j) = \sum_{i\in I}\sum_{n=0}^{K}\beta_{ij}\cdot\gamma_n\cdot\bar{f}_i\cdot x_{ij}\cdot P_{nj}(\lambda_j, y_j) = \sum_{i\in I}\beta_{ij}\cdot\bar{f}_i\cdot x_{ij}\sum_{n=0}^{K}\gamma_n\cdot P_{nj}(\lambda_j, y_j).$$

As λ_j increases, queues grow longer. Then, there is an index k such that, the sum of all probabilities $P_{nj}(\lambda_j)$ for all $n > k$ must increase, while the sum of the probabilities $P_{nj}(\lambda_j)$ for $n < k$ must decrease in the same amount, to keep $\sum_n P_{nj}(\lambda_j, y_j) = 1$.
Since $P_{kj} = 1 - \sum_{n<k} P_{nj}(\lambda_j, y_j) - \sum_{n>k} P_{nj}(\lambda_j, y_j)$,

$$g(\lambda_j, x_{ij}, y_j) = \sum_{i\in I}\beta_{ij}\cdot\bar{f}_i\cdot x_{ij}\left\{\gamma_k + \sum_{n<k}^{K}(\gamma_n-\gamma_k)P_{nj}(\lambda_j, y_j) + \sum_{n>k}^{K}(\gamma_n-\gamma_k)P_{nj}(\lambda_j, y_j)\right\}.$$

Also, since the demand is congestion averse, the factors $\gamma_n \leq 1$ decrease for growing values of n. Then, in the previous expression of $g(\lambda_j, x_{ij}, y_j)$, the factors $(\gamma_n - \gamma_k)$ multiplying the decreasing probabilities are positive, while the factors $(\gamma_n - \gamma_k)$ multiplying the increasing probabilities are negative. Consequently, $g(\lambda_j, x_{ij}, y_j)$ must be decreasing or, at least, non-increasing for increasing values of λ_j.

Furthermore, the function is bounded above by the maximum potential demand, and it is non-negative, *i.e.* bounded below by 0. Since λ_j is increasing from 0 to its maximum potential value, the function $F(\lambda_j) = g(\lambda_j, x_{ij}, y_j) - \lambda_j$ must have one and exactly one root. This root corresponds to the supply-demand equilibrium. □

Once the equilibrium equation is written, we can state the formulation.

$$\text{MAX} \sum_{j \in J} \lambda_j \tag{7}$$

$$
\begin{aligned}
s.t. \quad & \lambda_j = \sum_{i \in I} \sum_{n=0}^{K} \beta_{ij} \cdot \gamma_n \cdot \bar{f}_i \cdot x_{ij} \cdot P_{nj}(\lambda_j, y_j) \qquad && \forall j \in J && (8)\\
& \sum_{j \in J} x_{ij} \leq 1 && \forall i \in I && (9)\\
& x_{ij} \leq y_j && \forall i \in I, j \in J && (10)\\
& \sum_{j \in J} y_j = p && \forall j \in J && (11)\\
& x_{ij} \in 0, 1 && \forall i \in I, j \in J && (12)\\
& y_j \in Z && \forall j \in J && (13)
\end{aligned}
$$

where:

i,I = index and set of demand nodes.
j,J = index and set of candidates positions to locate facilities nodes.
λ_j = demand served by the facility at node j.
$\bar{f}_i$ = maximum demand originating at node i.
β_{ij} = demand elasticity factor with respect to travel from node i to node j.
γ_n = demand elasticity factor with respect to congestion.
P_{nj} = probability that there are n clients at the facility j
x_{ij} =1 if the node i is served by the facility at j 0 otherwise.
y_j = number of servers at node (facility) j.

The objective (7) maximizes the total expected demand over the system. Constraint (8) forces equilibrium at each node j. Constraint (9) states that the demand at node i is allocated to at most one facility. Constraint (10) prevents assigning demand to a facility that does not have a server. Constraint (11) limits the number of servers to be placed on the network to p. Constraints (12) and (13) define the integer nature of the variables.

Note that constraint (8) is a nonlinear function of λ_j. Given its structure, we see that it can be arranged as follows:

$$\lambda_j = \sum_{i \in I} \beta_{ij} \cdot \bar{f}_i \cdot x_{ij} \cdot \left[\sum_{n=0}^{K} \gamma_n \cdot P_{nj}(\lambda_j, y_j) \right]$$

or

$$\frac{\lambda_j}{\left[\sum_{n=0}^{K} \gamma_n \cdot P_{nj}(\lambda_j, y_j) \right]} = \sum_{i \in I} \beta_{ij} \cdot \bar{f}_i \cdot x_{ij}. \tag{14}$$

Let us call $G(\lambda_j)$ the left hand side of equation (14). Equation (14) can be rewritten as:

$$G(\lambda_j) = \sum_{i \in I} \beta_{ij} \cdot \bar{f}_i \cdot x_{ij}. \tag{15}$$

In the objective of the integer programming formulation, we could replace λ_j by:

$$\lambda_j = G^{-1}\left(\sum_{i \in I} \beta_{ij} \cdot \bar{f}_i \cdot x_{ij}\right).$$

However, finding this inverse function becomes virtually impossible given its combinatorial complexity and non-linearity. Since there are no known optimal techniques that can be applied to this problem, we solve it using a heuristic procedure.

4. Heuristic procedure

The model can be seen as a p-median problem with co-location of facilities (or servers), plus the added supply-demand equilibrium constraint, that acts as a dynamic capacity constraint. The p-median problem is known to be NP-hard [11] and, by reduction, so is the problem we solve in this paper. Thus, the use of a heuristic is recommended.

The model presented here has a very similar structure to the model in [15], where Lagrangean Relaxation was used, together with a procedure to find demand equilibrium. Lagrangean relaxation [8] is based on relaxation of constraints, whose violation is penalized in the objective. In the case of linear integer optimization problems, Lagrangean relaxation finds good bounds for the solution. When used together with other heuristics, good feasible solutions can be obtained. In [15], the use of Lagrangean relaxation allowed to separate the problem into simpler sub-problems. The supply-demand equilibrium was found at each step, using a heuristic procedure.

In spite of the similar structures of the model in [15] and the model in this paper, it is not possible to use the same heuristic. This is because in the case at hand, as opposed to [15], the probabilities P_n depend not only on the allocation variables x_{ij}, but also on the location variables y_j, making the objective non-separable. This is also the case if any other constraint is relaxed. The fact that the objective is non-separable, forces a very time-consuming multidimensional search over the possible solutions. Even if the procedure could be used, the bounds obtained are not tight enough for considering it [1]. Consequently, we use a different procedure, based on Heuristic Concentration.

Heuristic Concentration (HC), a successful method developed by Rosing and ReVelle [24], is a two-stage procedure that solves location problems with fixed numbers of facilities, each one with one server. In its first (construction) stage, a large number of good solutions are found by first locating randomly the servers and improving the locations by a One-Opt procedure [26]. The One-Opt or One-Exchange procedure simply relocates facilities one by one, looking for a better solution. All locations that are present in at least one of the solutions obtained in the first stage are saved in a Concentration Set (CS). During the second (improvement) phase, either a heuristic or an exact procedure are used to obtain the best solution, restricted to locations belonging to the Concentration Set.

In order to solve this problem, we propose a nested, iterative algorithm: the outer procedure (a modified HC that allows co-location of servers and uses a reduced second stage) locates servers, while at each step of the outer procedure, the inner procedure solves the demand-facility assignment problem, including finding the root of the equilibrium equation. Note that if we know where the servers are located, we can efficiently assign the customers to them, so to maximize served demand.

To explain the inner procedure, recall that it is applied once server locations, *i.e.* variables y_j are known. Then, assignment variables x_{ij} are only defined for nodes j where there is at least one server. The problem reduces to:

$$\text{MAX} \sum_{j \in J} \lambda_j \tag{16}$$

$$s.a. \sum_{i \in I} \beta_{ij} \cdot \bar{f}_i \cdot x_{ij} = G(\lambda_j) \quad \forall j \in J \tag{17}$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I \tag{18}$$

$$x_{ij} \in 0, 1 \quad \forall i \in I, j \in J. \tag{19}$$

This is a nonlinear, integer multiple-knapsack problem, with an added supply-demand equilibrium constraint. Again, there are no available commercial software packages that can solve this reduced problem and, to our knowledge, no known optimal solution methods.

Since this problem is still very difficult, we solve it heuristically. We first assume that there is no elasticity to queuing, and assign the demand using a regular knapsack heuristic that maximizes $\beta_{ij} \cdot \bar{f}_i$. In other words, assigns demand i to the closest facility j. We then find the correct equilibrium at each facility and compute the objective of this initial solution. In a second stage, we seek improvements of this solution, using a one-exchange heuristic.

Recall that the equilibrium value of λ_j is given by equation (8):

$$F(\lambda_j) = g(\lambda_j, x_{ij}, y_j) - \lambda_j = 0 \quad \forall j \in J. \tag{20}$$

This function is nonlinear and combinatorial. Since the equilibrium needs to be computed each time new locations and demand assignments are found, a fast procedure needs to be devised for this computation. Simple line search procedures are shown to be too slow to be considered in this case, so we chose a Newton Raphson method, that iterates as follows, until convergence, $|\lambda_j^{k+1} - \lambda_j^k| < \epsilon$:

$$\lambda_j^{k+1} = \lambda_j^k - \left(\frac{\mathrm{d}F}{\mathrm{d}\lambda_j} \right)^{-1} F(\lambda_j^k). \tag{21}$$

The derivative is estimated by computing two neighboring values of the function:

$$\lambda_j^{k+1} = \lambda_j^k - \left[\frac{F(\lambda_j^k + \delta) - F(\lambda_j^k - \delta)}{2\delta}\right]^{-1} F(\lambda_j^k + \delta).$$

Follows a flow description of the heuristics.

First (construction) stage

1. Locate randomly the p servers. Locate a facility wherever there is at least one server.
2. Assign demand to servers.
   a. Find for each demand i, the p facilities j with greatest values of $\beta_{ij} \cdot \bar{f}_i$.
   b. Assign each demand node to the facility with the greatest value of $\beta_{ij} \cdot \bar{f}_i$. Compute the supply-demand equilibrium by finding the root of function $F(\lambda_j)$.
   c. Use a One-Opt procedure to improve the assignment (*i.e.* reassign each demand, one by one, to all candidate facilities, computing each time the supply-demand equilibrium). For each demand node, use as candidate facilities only those in the list of the p facilities with greatest values of $\beta_{ij} \cdot \bar{f}_i$.
   d. Save the best solution.
3. If the solution is one of the best five solutions found so far, save it in the Concentration Set (CS).
4. Repeat *n_max* times steps 1 to 3.

Second (improvement) stage

5. Find the maximum number of servers present at a single location over all the five best solutions found in the first stage. Call it M. Over the CS saved in the first stage, find if there are server locations that are present in all five best solutions. Save these server locations in a Concentration Set Open (CSO), as done in Rosing *et al.* [25]. These server locations in the CSO will belong to the final solution, so they must be considered as located in the remainder of the procedure.
6. For each one of the five best solutions found in the first stage, save the locations that are not in CSO, in a set called Concentration Set Free (CSF), particular to that solution.
7. For each one of the five solutions in CS, perform a Two-Opt procedure using the servers and locations in the corresponding CSF, as follows:
   a. Let L be the set of locations in CSF. Assume that each location can contain up to M servers.
   b. Replace each location in CSF by M copies. Assume that each one of these M copies can house only up to 1 server. There is now an augmented set of locations, each one being a single-server location.
   c. Enumerate the servers. Starting from the solution found in the first stage, relocate the servers two-by-two, using all possible combinations over the augmented set of locations. Complete the cycle.

   d. After every relocation, assign demands to servers, as it was done in the first stage, and compute the value of the objective. Save the best solution obtained so far.
   e. Repeat the cycle of two-by-two relocations until the best solution does not change. Save the best solution.
8. Save the best solution as final.

The first stage of this heuristic requires p assignments (step 1), and n^2 products and comparisons (steps 2a and 2b). In step 2c, for the the One-Opt procedure every possible exchange of a server need to be considered. There are p servers and $(n-p)$ possible alternate locations, which makes $p(n-p)$ possible exchanges. For each exchange, p runs of Newton Raphson must be performed, as well as n comparisons (one for each demand) in order to know if the server that moved, is closer than the currently assigned server to that particular demand. The total number of operations is then of $O((n+pNR)(p(n-p))$, where NR is the time needed for convergence of Newton Raphson. Thus, step 2c is the most expensive one. If p is close to n, then this figure is $O(NRn^3)$, where NR is the time needed by the Newton Raphson procedure. This stage is repeated 200 times in our case. In the second stage, clearly the most expensive step is the Two-Opt procedure, which is of $O(NRn^4)$.

A broad range of values has been used by different authors for the number of repetitions of the first stage and the number of best solutions that are kept in the CS. Repetitions go from 25 [23] to 200 [24]. This parameter influences the quality of the CS; the more repetitions, the greater the probability that the "best" solutions are good. For the number of solutions in CS, [24] use a value of 5 while [21] experiments with values of 5, 10, and 15, and [22] chose 10.

As the number of solutions in CS is increased, the chances that the optimal solution (or at least a better solution) can be found in the second stage are better, because there are more options to choose from. However, a larger CS also increases the run time of the algorithm by increasing the number of eligible solutions. On the other hand, since solutions that are close to optimality have a greater chance of being similar to each other, the CS does not increase linearly with the number of repetitions of the first stage for a given number of solutions in CS, but it does so at a decreasing rate.

Having these facts into consideration, we chose the same values of repetitions and solutions in CS as in [24].

5. Numerical examples

As stated in Berman and Krass [2], this is a very hard class of problems, being all existing formulations suitable only for small instances. In other words, any realistic size problem has to be reduced, for example by demand aggregation, to smaller sizes. Consequently, we built two reduced size example networks shown in Figures 3 and 4. The parameters were selected to show the sensitivity of the solution to changes in the number of servers: the maximum queue length was set

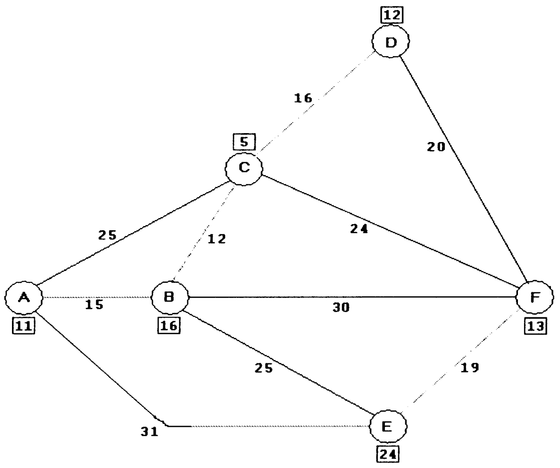

FIGURE 3. 6-node network.

at 7, that is, K – number of servers = 7, and the service rate of each server to 15. The factor β_{ij} was chosen to decrease linearly, reflecting the loss of 1% of the demand per each unit of distance. The factor γ_n decreases also linearly with the number of people in queue at the facility, n, in such a way that a 40% of people refuses service when the queue is 7 people long. In order to evaluate our procedure, we solved some instances using total enumeration, which is the only alternative method for the problem at hand, since no other methods have been proposed in the literature.

In order to take advantage of the random start, the heuristic was run ten times for each number of servers. As shown in Table 1, for all the instances we solved using enumeration (6-node network, p = 2, 3 and 5), the procedure found the optimal value. When the optimal value was not found by the procedure, the difference between the best and worse solutions was of at most of 2.42% (Tab. 1).

The algorithm was programmed in Visual BASIC, on a Pentium III, with 500 Mhz with 128 Mb RAM memory. The routines were not optimized for best running time.

The first column in Table 1 shows the number of the run and the second column the number of servers to locate. The next three columns show the results for the six-node network: the value of the objective; the difference in percentage with the optimal objective computed using total enumeration, or the best in the ten runs; and the running time in hours, minutes and seconds, respectively. The last three columns show the same figures for the 12-node network. Shown in italics, are those cases in which the procedure did not find the optimal or best value after 10 runs. Table 2 shows the server locations for the 12-node network.

Table 3 shows the execution times it took to perform total enumeration. As the table shows, these run times are considerably longer than the execution times

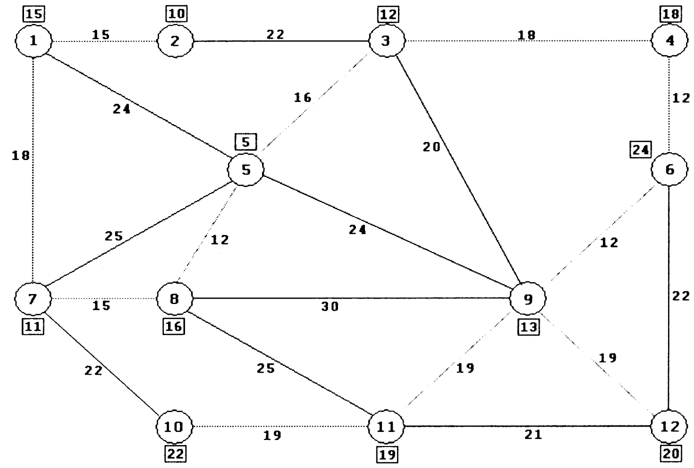

FIGURE 4. 12-node network.

required by the heuristic procedure. In some cases, total enumeration had to be stopped after very long times.

It is interesting to note that, looking at the five best solutions found at the first stage of the procedure, and the final solutions obtained by improving on each one of them during the second stage, not always the best first-stage solution leads to the best second-stage solution. An example can be seen in Table 4, for 12 nodes, 5, 10 and 15 servers. The Table shows which one of the best five solutions of stage 1, leads to the best final objective. This observation validates the choice of applying the second stage separately on each one of these five solutions, instead of using the regular HC procedure.

In all the optimal or best solutions obtained after running the entire procedure, the demands are assigned to one of the three closest facilities (*i.e.*, one of the three servers with higher values of the product). This fact is intuitively correct, since if there is no queuing at facilities, the assignment will be made to the closest facility. As the queues grow, the waiting time becomes more important than travel time and the assignment changes from the closest facility to the second closest, third closest, and so on. If waiting time or queues keep growing, and if all facilities have the same number of servers, the queues of all facilities become balanced, *i.e.* all queues have the same length, and the locations of servers or facilities become irrelevant. If facilities are equipped with different numbers of servers and waiting times are long enough, the waiting times at different facilities become balanced, the queue lengths are proportional to the number of servers and, again, the locations of the facilities are irrelevant. This effect is enhanced when the elasticity to waiting time is higher than the elasticity to travel time, as in our examples. The results show a rather balanced load over the system, being the difference in demand between the most congested center and the less congested one, at most 21.5%.

Table 1. Results for the test networks.

		6-node network			12-node network		
Run	p	Objective	Difference	Run time	Objective	Difference	Run time
1	2	38.221	0.00%	00:00:37	54.635	0.00%	00:02:12
2		38.221	0.00%	00:00:39	54.635	0.00%	00:02:30
3		38.221	0.00%	00:00:38	54.635	0.00%	00:02:36
4		38.221	0.00%	00:00:41	54.635	0.00%	00:02:10
5		38.221	0.00%	00:00:37	54.635	0.00%	00:02:32
6		38.221	0.00%	00:00:39	54.635	0.00%	00:02:14
7		38.221	0.00%	00:00:37	54.635	0.00%	00:02:07
8		38.221	0.00%	00:00:35	54.635	0.00%	00:02:10
9		38.221	0.00%	00:00:32	54.635	0.00%	00:02:05
10		38.221	0.00%	00:00:31	54.635	0.00%	00:01:21
1	3	47.755	0.00%	00:02:01	*69.252*	*0.29%*	00:07:38
2		47.755	0.00%	00:01:45	*69.252*	*0.29%*	00:11:19
3		47.755	0.00%	00:01:24	69.456	0.00%	00:12:05
4		47.755	0.00%	00:01:46	*69.252*	*0.29%*	00:11:06
5		47.755	0.00%	00:01:46	*69.252*	*0.29%*	00:10:24
6		47.755	0.00%	00:02:08	*69.252*	*0.29%*	00:04:44
7		47.755	0.00%	00:01:43	*69.146*	*0.45%*	00:07:47
8		47.755	0.00%	00:01:43	69.456	0.00%	00:06:43
9		47.755	0.00%	00:01:43	*69.252*	*0.29%*	00:04:48
10		47.755	0.00%	00:02:08	*69.252*	*0.29%*	00:04:42
1	5	63.102	0.00%	00:02:50	94.066	0.00%	01:13:50
2		63.102	0.00%	00:03:25	94.066	0.00%	01:31:18
3		63.102	0.00%	00:04:33	94.066	0.00%	01:06:32
4		63.102	0.00%	00:05:54	*94.006*	*0.06%*	01:13:31
5		63.102	0.00%	00:06:19	*94.006*	*0.06%*	00:58:45
6		63.102	0.00%	00:04:26	*94.006*	*0.06%*	01:08:58
7		63.102	0.00%	00:03:49	*93.926*	*0.15%*	01:05:19
8		63.102	0.00%	00:06:45	94.066	0.00%	00:50:36
9		63.102	0.00%	00:04:23	*93.681*	*0.41%*	01:32:17
10		63.102	0.00%	00:03:33	94.066	0.00%	01:09:14
1	10	77.13	0.00%	00:10:42	136.898	0.00%	07:44:56
2		77.13	0.00%	00:12:14	136.898	0.00%	05:18:54
3		77.13	0.00%	00:05:27	*136.571*	*0.24%*	11:32:08
4		77.13	0.00%	00:10:44	136.898	0.00%	09:01:25
5		77.13	0.00%	00:14:17	136.898	0.00%	07:32:24
6		77.13	0.00%	00:08:50	136.898	0.00%	08:56:38
7		77.13	0.00%	00:11:07	*136.389*	*0.37%*	05:23:17
8		*76.75*	*0.49%*	00:09:51	136.898	0.00%	07:58:06
9		77.13	0.00%	00:07:56	*136.572*	*0.24%*	11:55:54
10		77.13	0.00%	00:12:25	136.898	0.00%	11:07:54
1	15	80.362	0.00%	00:16:14	*160.565*	*0.05%*	18:14:24
2		80.362	0.00%	00:14:18	*160.412*	*0.14%*	23:42:45
3		80.362	0.00%	00:12:42	160.643	0.00%	21:09:45
4		*80.133*	*0.29%*	00:10:12	*160.446*	*0.12%*	20:14:54
5		*80.133*	*0.29%*	00:08:19	*157.71*	*1.83%*	19:57:20
6		*80.149*	*0.26%*	00:08:31	160.643	0.00%	20:40:03
7		*80.149*	*0.26%*	00:08:30	*160.023*	*0.39%*	21:08:50
8		80.362	0.00%	00:12:45	*156.749*	*2.42%*	19:42:25
9		80.362	0.00%	00:11:45	*160.17*	*0.29%*	20:52:13
10		80.362	0.00%	00:17:54	*159.927*	*0.45%*	24:21:44

Table 2. Server locations for the 12-node network.

p	Nodes	Assignment	Percentage served of total demand assigned
2	G, I	G: A, B, E, G, H, J I: C, D, F, I, K, L	G: 31.98 I: 27.71% Total: 29.53%
3	G, I, I	G: A, B, G, H, J I: C, D, E, F, I, K, L	G: 33.21% I: 40.44% Total: 37.54%
5	B, H, J, F, F	B: A, B, C F: D, F, I, L H: E, G, H J: J, K	B: 49.34% F: 51.46% H: 54.99% J: 47.85% Total: 50.85%
10	A, A, C, D, F, F, H, J, K, L	A: A, B, G C: C, E D: D F: F, I H: H J: J K: K L: L	A: 74.11% C: 74.58% D: 75.06% F: 75.82% H: 78.62% J: 68.68% K: 73.36% L: 71.73% Total: 74.00%
15	A, A, C, D, F, F, H, H, I, J, J, K, K, L, L	A: A, B C: C D: D F: F H: E, G, H I: I J: J K: K L: L	A: 85.80% C: 86.08% D: 75.06% F: 90.79% H: 78.92% I: 84.23% J: 92.69% K: 95.19% L: 94.41% Total: 86.83%

Table 3. Execution times for complete enumeration.

p	Run time – 6-nodes	Run time – 12-nodes
2	00:31:03	89 hr*
3	01:52:36	undetermined
5	11:05:00	undetermined
10	180 hr*	undetermined
15	108 hr*	undetermined

* Unfinished runs.

Due to the maximum queue length and the average service rate, that are the same for the 6-node and the 12-node networks, a smaller percentage of the demand is covered in the last one, for the same number of servers. Figure 5 shows the trade-off curve of demand covered versus number of servers. As expected, the

Table 4. Solutions of the first stage leading to the best objective.

Run	p	Objective	First stage solution leading to the best objective
1	5	94.066	5
2		94.066	1
3		94.066	5
4		*94.006*	1
5		*94.006*	1
6		*94.006*	2
7		*93.926*	2
8		94.066	5
9		*93.681*	5
10		94.066	5
1	10	136.898	1
2		136.898	3
3		*136.571*	4
4		136.898	1
5		136.898	1
6		136.898	5
7		*136.389*	4
8		136.898	5
9		*136.572*	4
10		136.898	1
1	15	*160.565*	5
2		*160.412*	4
3		160.643	1
4		*160.446*	3
5		*157.710*	1
6		160.643	2
7		*160.023*	1
8		*156.749*	1
9		*160.170*	1
10		*159.927*	4

contribution of a $(p+1)$th server to demand coverage decreases with increasing values of p.

6. Conclusions

We present a new model and procedure for solving the server allocation problem in the case of facilities subject to queuing effects, and when demand for the services provided is elastic to both travel time to the facilities and queue lengths at the facilities. The procedure is intended for long term, strategic purposes.

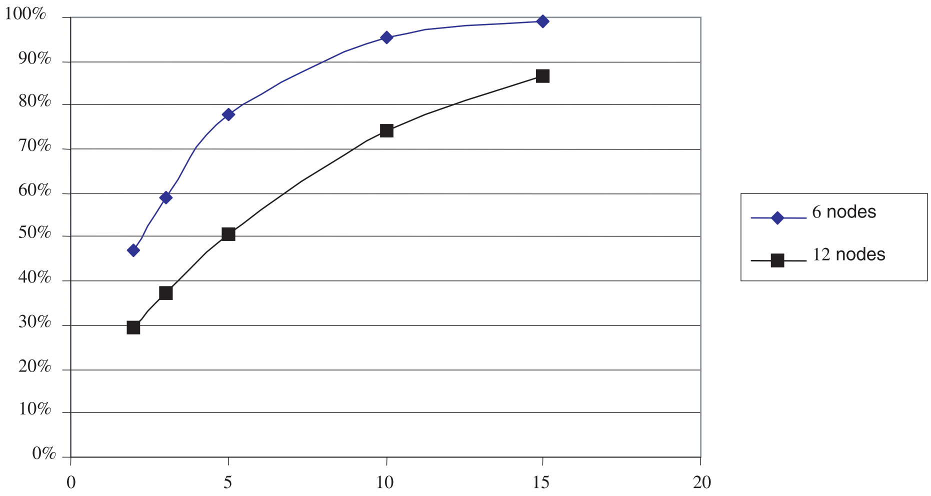

Figure 5. Percentage of total demand covered versus number of servers.

The model is highly nonlinear, with an equilibrium equation, and there are no known methods to solve this class of problems in the literature. We propose a heuristic procedure that provides a solution in reasonable times, as compared to enumeration. The procedure has a resemblance to Rosing and ReVelle's Heuristic Concentration [24], which had to be modified in order to reduce the running times and allow co-location of servers. It uses Newton-Raphson techniques for finding the demand equilibrium and a Two-Opt improvement heuristic applied on several starting solutions. The solutions found for small examples are optimal, while the solutions found for larger examples, using several runs of the heuristic, have a small dispersion, in spite of the random starting points of each run. This could indicate that the heuristic is finding solutions that are close to the optimal value.

In practice, the only data needed to use the model and procedure, are the demand elasticity parameters β_{ij} and γ_n that can be found by polls, focus groups or similar techniques. No analytical curves are needed.

For small networks and small numbers of servers, the procedure finds optimal values. It has not been possible to find optimal values of the solutions for larger instances, due to the long time it takes for total enumeration.

Future research is needed in order to find more efficient heuristics for larger problems.

References

[1] F.J. Barros, *Asignacion de Recursos en una Red Congestionada y con Demanda Elastica.* Unpublished Master's Thesis, Pontificia Universidad Catolica de Chile (2004).

[2] O. Berman and D. Krass, Facility Location Problems with Stochastic demands and Congestion, in *Facility Location: Applications and Theory*, edited by Z. Drezner and H.W. Hamacher. Springer-Verlag, New York (2002) 329–371.

[3] O. Berman and R. Larson, Optimal 2-Facility Network Districting in the Presence of Queuing. *Transportation Science* **19** (1985) 261–277.

[4] O. Berman and S. Vasudeva, *Approximating Performance Measures for Public Services*, working paper, Joseph L. Rotman School of Management, University of Toronto, Canada (2000).

[5] O. Berman, R. Larson and C. Parkan, The Stochastic Queue P-Median Location Problem. *Transportation Science* **21** (1987) 207–216.

[6] O. Berman and R. Mandowsky, Location-Allocation on Congested Networks. *Eur. J. Oper. Res.* **26** (1986) 238–250.

[7] M. Brandeau, S. Chiu, S. Kumar and T. Grossman, Location with Market Externalities, in *Facility Location: A Survey of Applications and Methods*, edited by Z. Drezner. Springer-Verlag, New York (1995) 121–150.

[8] M. Daskin, *Networks and Discrete Location: Models, Algorithms and Applications.* Wiley-Interscience Series in discrete Mathematics and Optimization, John Wiley (1995).

[9] L. Hakimi, Optimal Location of Switching Centers and the absolute centers and medians of a graph. *Oper. Res.* **12** (1964) 450–459.

[10] F. Hillier and G. Lieberman, *Introduction to Operations Research.* Holden-Day Inc., Oakland, CA (1986).

[11] O. Kariv and L. Hakimi, An algorithmic approach to network location problems, part ii: The p-medians. *SIAM J. Appl. Math.* **37** (1979) 539–560.

[12] G. Laporte, F. Louveaux and L. Van Hamme, Exact solution to a location problem with stochastic demands. *Transportation Science* **28** (1994) 95–103.

[13] V. Marianov and C. ReVelle, The standard response fire protection siting problem, INFOR: The Canadian *J. Oper. Res.* **29** (1991) 116–129.

[14] V. Marianov and C. ReVelle, The capacitated standard response fire protection siting problem: deterministic and probabilistic models. *Ann. Oper. Res.* **40** (1992) 303–322.

[15] V. Marianov, Location of Multiple-Server Congestible Facilities for Maximizing Expected Demand, when Services are Non-Essential. *Ann. Oper. Res.* **123** (2003) 125–141.

[16] V. Marianov and D. Serra, Location Problems in the Public Sector, in *Facility Location: Applications and Theory*, edited by Z. Drezner and H.W. Hamacher. Springer-Verlag, New York (2002) 119–150.

[17] V. Marianov and D. Serra, Location-Allocation of Multiple-Server Service Centers with Constrained Queues or Waiting Times. *Ann. Oper. Res.* **111** (2002) 35–50.

[18] V. Marianov and D. Serra, Location models for airline hubs behaving as M/D/c queues. *Comput. Oper. Res.* **30** (2003) 983–1003.

[19] V. Marianov and D. Serra, Location of Multiple-Server Common Service Centers or Public Facilities, for Minimizing General Congestion and Travel Cost Functions. Research Report, Department of Electrical Engineering, Pontificia Universidad Catolica de Chile.

[20] C. ReVelle and R. Swain, Central Facility Location. *Geographical Analysis* **2** (1970) 30–42.

[21] K. Rosing, Heuristic concentration: A study of stage one, Tinbergen Institute Discussion Papers. Tinbergen Institute, Amsterdam/Rotterdam (1998).

[22] K. Rosing, Heuristic concentration: a study of stage one. *Environment and Planning B* **27** (2000) 137–150.

[23] K. Rosing and M.J. Hodgson, Heuristic concentration for the p-median: an example demonstrating how and why it works. *Comput. Oper. Res.* **29** (2002) 1317–1330.

[24] K. Rosing and C. ReVelle, Heuristic Concentration: Two stage solution construction. *Eur. J. Oper. Res.* **97** (1997) 75–86.

[25] K. Rosing, C. ReVelle and D. Schilling, A Gamma Heuristic for the P-Median. *Eur. J. Oper. Res.* **117** (1999) 522–532.

[26] M. Teitz and P. Bart, Heuristic methods for estimating the generalized vertex median on a weighted graph. *Oper. Res.* **16** (1968) 955–965.

[27] J. Zhou and L. Baoding, New stochastic models for capacitated location-allocation problem. *Comput. Industrial Eng.* **45** (2003) 111–125.