LOCATION OF MULTIPLE SERVER COMMON SERVICE CENTERS OR PUBLIC FACILITIES FOR MINIMIZING GENERAL CONGESTION AND TRAVEL COST FUNCTIONS

V. Marianov[1], D. Serra[2]
[1]Department of Electrical Engineering
Pontificia Universidad Católica de Chile, Santiago, CHILE
e-mail: *marianov@ing.puc.cl*
[2]Department of Economics and Business, and GREL-IET
Universitat Pompeu Fabra, Barcelona, SPAIN
e-mail:*daniel.serra@econ.upf.es*

ABSTRACT

We propose a model and solution methods, for locating a fixed number of multiple-server, congestible common service centers or congestible public facilities. Locations are chosen so to minimize consumers' congestion (or queuing) and travel costs, considering that all the demand must be served. Customers choose the facilities to which they travel in order to receive service at minimum travel and congestion cost. As a proxy for this criterion, total travel and waiting costs are minimized. The travel cost is a general function of the origin and destination of the demand, while the congestion cost is a general function of the number of customers in queue at the facilities.

Keywords: Location modelling, congestion costs
AMS Classification: 90B22, 90B15, 90B80
JEL Classification: C61, J80

1. Introduction and literature review

Some large organizations, with geographically distributed or extensive facilities, have internal clients (workers) that must receive some service at centers belonging to the same company. These centers must be located in such a way that people working at any facility can use them; workers have to travel from their workplaces to these centers and wait until served. In this case, the companies may want to minimize the cost associated with the time spent by their workers both traveling to, and waiting in lines at these centers. An instance of this setting is a large university campus with many buildings, where food or similar services need to be located to serve all workers and students. The same situation occurs when there are common service centers established as joint ventures by their users. Examples are the co-op stores; the users want to have the centers as close as possible to them, and to receive a fast service. Otherwise, the benefit of having established the common centers or stores may not be clear to some of them. More examples can be found among public services, electrical utilities, phone companies, and so on, with geographically distributed offices. Although the total demand for attention at these offices does not depend on the locations or number of service centers, the providers may wish to promote a good image, by minimizing customers' inconvenience (or cost) related to travel and waiting time. In turn, each individual user or customer minimizes her/his own travel and waiting costs when seeking service. Assuming the customers of the services under study are a small part of all the people traveling over the network, travel costs depend only on the distance or the time it takes to the customer to go from her/his starting point to the center she/he chooses. However, waiting costs depend jointly on the decisions made by all customers. In fact, the more customers decide to patronize some center, the more congested it becomes, so the waiting time of all the customers using this center increases. Furthermore, the cost that customers assign to the time they spend traveling may be very different to the cost they assign to the time spent waiting at the center. Also, different types of services can have different cost functions associated to travel and waiting. Finally, different customers could potentially have different perceptions of the cost of time (cost functions).

All these facts must be taken into account when the company, service or group of users have to decide on how many centers to open, where to locate them, and how many servers (clerks, attendants, cashiers, machines) to allocate to each one of them, so to minimize the travel and waiting cost functions. We propose a method for making these decisions.

A very good review of published models, as well as some additional models for location on congested networks, is presented in Brandeau *et al* (1995). The models cover several types of demand (elastic or inelastic to congestion and travel time) and assignment of customers to facilities (either forced by a central planner or made by user choice). In most cases of inelastic demand, the models locate one, or at most two facilities. In all cases, additional to travel distance or time, the queuing delay is the main issue, as opposed to the number of customers in the system. As we will show later, formulas of queuing delay are more difficult to handle than formulas of number of customers in the system, although delay and number of customers are related and consequently, are equivalent.

Lately, additional models for location of congested facilities have been published. Marianov and Serra (1998, 2001), rather than minimizing waiting time, choose to use a constraint that forces the probability of a queue longer than a certain number, to be less than a predefined value. Jamil *et al* (1999) locate a single server, as opposed to several, multiple-server facilities. In the area of telecommunications, many approaches have appeared in the literature for the design of networks so as to either minimize communications delay or to keep this delay constrained to some extent (Rolland *et al*, 1999). Although these models are similar to those used for facility location, again, they consider queuing delay in their formulations.

The model we propose optimizes the location of a fixed number of service centers, each with several servers, in such a way that the total travel and waiting costs are minimized, as two different objectives. The number of servers per center is fixed, but can be determined by repeatedly solving the model. As opposed to existing models, in the model we present, instead of minimizing the travel and expected waiting *time*, we minimize separately the travel *cost* and the expected waiting *cost*, computed as an expected cost, averaged over the number of customers in the system. We also allow the possibility that *for customers coming from different origins, the cost function might be* different (although we do not provide a solution method for this case); thus, we can take into account the *differences in customers' perceptions* when optimizing the quality of service of the whole system. Furthermore, we provide formulations that *allow us to use general cost functions,* which in the case of congestion is done through the use of queue lengths instead of waiting times, being both equally meaningful measures of congestion.

If both cost objectives are measured in monetary terms, they can be added to obtain a single objective. Nevertheless, if their units are different, for example, travel and waiting times instead of travel and waiting costs, which can be perceived differently by customers, then a multiobjective approach has to be chosen. Furthermore, one of the objectives is nonlinear.

We present the model together with *two solution procedures*. The first procedure solves the problem if objectives are cast in monetary terms, or if the weighting method is chosen for the multi-objective optimization formulation. This procedure uses a metaheuristic based on tabu search and ant-colonies to solve the problem.

The second procedure solves the multi-objective, nonlinear optimization model, using the constraint method (Cohon, 1978). In this method, one of the objectives is kept as such, while the remaining objectives are constrained to have acceptable values. These acceptable values or right hand sides of the new constraints, take different values in successive runs, so that trade-off curves can be obtained. In our case, by transforming the nonlinear objective into a constraint, we are also able to linearize the problem and solve it using commercial integer programming software, provided some simplifying assumptions are made.

The paper is structured as follows. We first develop the nonlinear multiobjective formulation of the problem. Then we present the metaheuristic which is useful when the weighting method is chosen, or when both objectives are measured in monetary terms and can be added to form a single objective. We offer computational experience on a set of test instances of the problem. Then, we present the constraint method, and provide some computational experience on a 55-node network using CPLEX. We do not use the same instances of the problem for both methods, because in this way the reader can see different interesting aspects of the problem and the methods.

2. Development of the multiple-server model

The formulation is cast using *p*–median constraints. The model locates *p* facilities, each with *s* servers, and determines the allocation of the demands to each facility, so that the travel and waiting cost functions are minimized, as follows:

$$\begin{array}{lll}
\text{Min} & [Z_1, Z_2] & \\
 & Z_1 = \text{ (Travel Cost)} & \\
 & Z_2 = \text{(Expected Congestion Cost)} & (1)\\
\text{s. t.} & \sum_j x_{ij} = 1 \quad \forall\, i & (2)\\
 & x_{ij} \le y_j \quad \forall\, i, j & (3)\\
 & \sum_j y_j = p & (4)\\
 & x_{ij} \in \{0,1\} \qquad \forall i,j & (5)\\
 & y_j \in \{0,1\} \qquad \forall j &
\end{array}$$

In this model, variable x_{ij} is one if demand *i* is allocated to facility *j*, and zero otherwise. Variable y_j is one when a facility is located at candidate site *j*, and zero otherwise. The objectives to be minimized are both costs. Constraint (2) forces all demands to be allocated to exactly one facility. Constraint (3) states that a demand cannot be served from a site without a facility. Constraint (4) forces *p* facilities to be located, and constraint (5) ensures integrality of the variables.
We now propose expressions for the cost functions.

Travel Cost Function

Finding an expression for a general travel cost function is done in the same way as in the *p*-median-like models. The travel cost per customer is a function of the customer's origin (demand node) and destination (facility), or travel distance, or travel time. This function can take any form. It could be nonlinear, discontinuous and non-differentiable. It could be even a different function for each pair origin – destination. Each origin (demand node) *i* is known, and for each origin there are a finite number of destinations *j*. Thus, the value c_{ij} of this general function can be computed for each pair origin-destination (*i, j*) before solving the optimization problem, and this value c_{ij} used in the first objective as follows:

$$Z_1 = \sum_i \sum_j h_i c_{ij} x_{ij} \tag{6}$$

where c_{ij} is the cost (per customer) of traveling from *i* to *j*, and h_i is the demand rate at demand node *i*.

<u>*Congestion Costs*</u>

In most of the existing models, congestion is expressed in terms of the expected waiting times (or delays). In order to obtain a tractable mathematical function representing a general cost function to be minimized, we use the number of customers in the system instead, and minimize directly an expected cost. Recall that the expected waiting time and the expected number of customers in the system are related through Little's law, so both are equivalent. In order to compute the expected congestion cost, we assume a finite queue multiple servers M/M/s/K queuing system at each facility. Calls arrive to each facility at a rate λ_j, given by:

$$\lambda_j = \sum_i h_i x_{ij} \tag{7}$$

and are served at a rate μ_j. Each state *n* of the queuing system, representing *n* customers in the system, has a probability $P_n(\lambda_j)$, which depends on the arrival rate λ_j. Equivalently, if $\rho_j = \lambda_j / \mu_j \leq s$, and *s* is the number of servers at *j*, the probability can be expressed as $P_n(\rho_j)$, according to the following equations (Hillier and Lieberman, 1986), where the subscript *j* has been omitted for simplicity purposes:

$$P_0(\rho) = \left[1 + \sum_{n=1}^{s} \frac{\rho^n}{n!} + \frac{\rho^s}{s!} \sum_{n=s+1}^{K} \left(\frac{\rho}{s} \right)^{n-s} \right]^{-1} \tag{8}$$

$$P_n(\rho) = \begin{cases} \dfrac{\rho^n}{n!} P_0(\rho) & \text{for } n \leq s \\ \dfrac{\rho^n}{s!s^{n-s}} P_0(\rho) & \text{for } s \leq n \leq K \\ 0 & \text{for } n \geq K \end{cases} \tag{9}$$

One set of similar equations can be written for each facility *j*. Instead of using explicitly a function of the waiting time in the objective, we assume that for a customer originating at node *i*, her/his congestion cost function (possibly different for each *i*) is discrete, and depends on the number *n* of customers (including him/herself) at the facility she/he chooses. This value is b_{in}. Using this value and the probabilities just defined, the expected queuing cost at a facility *j* for a customer originating at *i* is:

$$\sum_{n=0}^{K} b_{in} P_n(\lambda_j) .$$

For all the customers patronizing the facility *j*, the total cost is:

$$\sum_{i} h_i x_{ij} \sum_{n=0}^{K} b_{in} P_n(\lambda_j) = \sum_{i} \sum_{n=0}^{K} h_i x_{ij} b_{in} P_n(\sum_{i} h_i x_{ij})$$

Then, the objectives of the model are:

$$\mathrm{Z}_1 = \sum_{i} \sum_{j} h_i c_{ij} x_{ij} \quad \text{and}$$

$$\mathrm{Z}_2 = \sum_{j} \sum_{i} \sum_{n=0}^{K} h_i x_{ij} b_{in} P_n(\sum_{i} h_i x_{ij}) \qquad (10)$$

Adding constraints (2) to (5) completes the model. Note that objective Z_2 is the sum of the costs over all facilities *j*. The second objective is nonlinear.

3. A metaheuristic for the non-linear integer formulation

The model presented in the previous section is a combinatorial optimization problem. Many combinatorial problems are intractable and belong to the class of NP-Hard problems. Kariv and Hakimi (1979) prove that the p-Median problem is an NP-Hard problem on a general graph. Moreover, in this case, the inclusion of a non-linear additional objective reinforces the difficulty of solving the problem optimally.

The common belief in this field is that no efficient algorithm could ever be found to solve these inherently hard problems. Heuristics, and recently metaheuristics, are considered one of the choice methods for solving hard combination optimization problems. We propose the use of a metaheuristic procedure based on the MAX-MIN ant system together with TABU search for finding solutions to the proposed formulation.

The Ant System introduced by Colorni *et al* (1991a, 1991b), Dorigo *et al* (1996), Dorigo and Di Caro (1999), is a cooperative search algorithm inspired by the behavior of real ants. Ants lay down some quantity of an aromatic substance, known as pheromone, on their way to food. The likelihood with which an ant chooses a specific path, is correlated with the intensity of the pheromone on that path. The pheromone trail evaporates over time if no more pheromone is laid down by other ants, therefore the best paths have more intense pheromone and higher probability of being chosen. The Ant System approach associates pheromone trails to features of the solutions of a combinatorial problem, and can be seen as a kind of adaptive memory of previous solutions. Solutions are iteratively constructed in a randomized heuristic fashion biased by the pheromone trails left by the previous ants. The pheromone trails, τ_j, are updated after the construction of a solution, ensuring that the best features will have a more intense pheromone.

Recently, Stützle (1997) have proposed an improved version of the Ant System, denominated MAX-MIN Ant System. The MAX-MIN Ant system differs from the Ant System in the following way: only the best ant updates the trails in every cycle. To avoid stagnation of the search, i.e. ants always choosing the same path, Stüzle (1998a) proposed a lower and upper limit to the pheromone trail, τ_{min} and τ_{max}, respectively. Stützle and Hoos (1999), Stützle (1997,1998a) applied this procedure to the Traveling Salesman Problem, The Quadratic Assignment Problem and The Flow-Shop Scheduling Problem; and Lourenço and Serra (2000) applied it to the Generalized Assignment Problem.

Tabu Search is a metaheuristic that guides local heuristic search procedures to explore the solution space beyond local optimality. It was introduced by Glover (1989, 1990) and, in essence, Tabu Search explores a part of the solution space by repeatedly examining all neighborhoods of the current solution, and moving to the best neighborhood even if this leads to a deterioration of the objective function. This approach tries to avoid being trapped in a local optimum. In order to avoid the cycling back to a solution that has recently been examined, solutions are inserted in a tabu list that is constantly updated. Additionally, several criteria of flexibility can be used in the tabu search including aspiration and diversification.

The aspiration criterion is used as an insurance against restricting moves which would have led to finding high quality solutions. In other words, the aspiration criteria determines when a move to a new solution is done, even if tabu. Usually, this criterion states that if a move produces a feasible solution that is better than the best known solution, then the tabu status is disregarded and the move is executed. The diversification criterion is utilized to escape from local optima and is achieved by using a long - term memory function. It allows a broader exploration of the solution space by starting from solutions that have not been well explored. A more detailed description can be found in Appendix 1.

This method has been successfully applied to a wide variety of location problems: p-hub Location Problems (Klincewicz, 1992) and (Marianov *et al*, 1999), (r | *Xp)*- Medianoid and (r | *p*)- Centroid Problems (Benati and Laporte, 1994), the Vehicle Routing Problem (Gendreau *et al*, 1994) and p-Median Problem (Rolland *et al*, 1996).

In order to use the MAX-MIN ant system, we define the pheromone trail τ_j as the desirability of locating a center in j. The higher τ_j is, the more desirable is the location of a center in that node. The τ_j are initially set as

$$\tau_j = \left[\sum_i h_i c_{ij} \right]^{-1},$$

that is the inverse of the weighted travel cost from all demand nodes to the facility. In essence, once the initial τ_j are computed, the procedure has the following steps:

1. Construction of an initial solution. First, the nodes are ordered with respect to the probability function defined by
$$p_j = \frac{\tau_j}{\sum_{l \in J} \tau_l}.$$
The initial solution is chosen randomly, taking into account the probability distribution previously defined.
2. Solutions in the neighborhood of the initial solution are searched in an attempt to improve the initial solution. Tabu search is applied.
3. The pheromone trails are updated using the current solution in the following way: $\tau_j^{new} = \rho \tau_j^{old} + \Delta \tau_j$, where ρ ($0 < \rho < 1$) is the persistence of the trail, i.e. 1 - ρ, represents the evaporation. The updated amount is
$$\Delta \tau_j = \begin{cases} \tau_{\max} Q & \text{If an outlet is located at j} \\ 0, & \text{otherwise} \end{cases}$$
where Q is a preset parameter ($Q < 1$).
4. In the final stage, the MAX-MIN limits are checked and imposed $\tau_{min} \leq \tau_j \leq \tau_{max}$ $\forall j \in J$, if the updated pheromone falls outside the interval.

The termination condition of this iterative procedure is the number of total iterations.
A formal description of the metaheuristic procedure is presented in Appendix 1.

4. Computational Experience with the metaheuristic procedure

The non-linear formulation has been tested as follows: several random networks with 30 and 50 nodes (n) were generated. For each network, the demand rate at each node was randomly set between 100 and 500. Congestion costs were set equal for all customers regardless of location. If j represents the number of people in queue, these cost where set as follows: $c_j = s/10$ for $j \leq s$ and $c_j = c\ (j - 1) + 1/10$ for $j > s$. Also, for each network, the number of facilities (p) to be located was set to 3, 4 and 5 and for each n and p the number of servers (s) was set to 2, 3 and 4. The service rate was set equal to

$$\frac{fact * calltot}{p}$$

where *calltot* is the total number of calls in the system, p the number of facilities to be located and *fact* is a preset factor equal to 2, 3 and 4.

For each *n, p, s* and *fact*, 25 networks were generated, giving a total of 1,350 instances of the metaheuristic. The maximum number of people in the queue was set to 10.

For each instance of the metaheuristic, the number of iterations in the Max-min ant-colony procedure was set to 50. For each instance of the tabu search heuristic, the number of one-opt trades in the tabu phase of the heuristic was set to 40. The diversification phase was executed three times. The size of the neighborhood (the *m* nodes around the current solution) was set between 4 and 8. The size of the tabu list was set to 6. The values of the parameters of the metaheuristic were set to $Q = 0.10$, $\rho = 0.75$, $\tau_{\max} = 2p * \max(\tau_j)$ and $\tau_{\min} = (1/2p) * \min(\tau_j)$ (where, p is the number of centers to locate).

In order to check the results obtained by the metaheuristic, the 1,350 instances were also solved using complete enumeration. Results are presented in Tables 1 and 2. For both tables, the first column (NP) represents de number of facilities to be located. In the second column (S) the number of servers is listed. The third column indicates the preset factor used to compute the service rate. The fourth and fifth colums represent de number of instances were the algorithm did not obtain optimal solutions in total and percentage terms respectively. In the sixth and seventh columns the average and maximum deviation from optimality is presented. Finally, the last column shows the average computer times in seconds.

For n = 30, 94.6 % of the 675 results obtained with the metaheuristic were optimal. The average deviation from optimality was equal to 0.97% and the maximum deviation (4.10%) was obtained when p = 2, s = 4, and *fact* = 3. The average computing time ranged between 4.79 and 10.42 seconds on a Pentium III with 256Mb of RAM.

For n = 50, 94.5% of the 675 results obtained with the metaheuristic were optimal. The average deviation from optimality was equal to 1,46% and the maximum deviation (9.63%) was obtained when p = 2, s =3, and *fact* = 2. The average computing time ranged between 7.61 and 16.77 seconds.

NP	S	FAC	Nonoptimal solutions	% non optimal	Average deviation	Maximum deviation	Average time
3	2	2	0	1.3%	-	-	4.77
		3	0		-	-	4.74
		4	1		0.41%	0.41%	4.69
	3	2	2	4.0%	1.37%	1.85%	4.87
		3	1		2.35%	2.35%	4.83
		4	0		-	-	4.76
	4	2	1	1.3%	1.52%	1.52%	4.75
		3	0		-	-	4.75
		4	0		-	-	4.88
4	2	2	3	8.0%	1.69%	3.17%	7.18
		3	2		1.41%	2.88%	7.17
		4	1		0.63%	0.63%	7.26
	3	2	3	8.0%	1.42%	3.24%	7.27
		3	2		1.14%	3.00%	7.50
		4	1		0.97%	1.02%	7.49
	4	2	2	6.6%	1.32%	3.84%	7.38
		3	1		0.13%	1.13%	7.26
		4	2		1.72%	3.74%	7.30
5	2	2	0	6.6%	1.85%	2.21%	10.17
		3	2		1.19%	2.97%	9.93
		4	3		1.80%	4.10%	9.95
	3	2	1	5.3%	1.12%	1.66%	10.38
		3	1		0.65%	0.90%	10.34
		4	2		1.07%	2.28%	10.34
	4	2	1	6.0%	0.15%	0.39%	10.39
		3	2		1.37%	2.09%	10.42
		4	3		1.16%	3.65%	10.17

Table 1: Results with *n* = 30

NP	S	FAC	Nonoptimal solutions	% non optimal	Average deviation	Maximum deviation	Average time
3	2	2	1	5.3%	0.25%	0.25%	7.67
		3	2		1.26%	2.44%	7.61
		4	0		-	-	7.67
	3	2	0	4.0%	-	-	7.76
		3	1		0.35%	0.35%	7.80
		4	2		1.28%	2.09%	7.72
	4	2	0	2.6%	-	-	7.87
		3	0		-	-	7.70
		4	2		1.54%	1.92%	7.80
4	2	2	1	8.0%	1.07%	3.86%	11.52
		3	2		1.59%	5.17%	11.65
		4	3		1.95%	3.19%	11.57
	3	2	1	6.6%	1.62%	1.82%	11.82
		3	1		2.67%	6.66%	11.92
		4	3		1.32%	3.23%	11.92
	4	2	2	10.6%	1.00%	2.92%	11.84
		3	3		2.39%	2.88%	11.96
		4	3		0.21%	0.36%	11.70
5	2	2	4	9.3%	2.09%	4.48%	16.45
		3	2		4.86%	9.63%	16.44
		4	1		3.99%	6.70%	16.50
	3	2	3	14.6%	1.12%	2.36%	16.77
		3	4		1.96%	5.47%	16.55
		4	5		1.50%	4.62%	13.75
	4	2	2	13.3%	1.89%	3.78%	13.88
		3	4		2.02%	5.03%	13.86
		4	4		2.57%	6.56%	13.96

Table 2 : Results with *n* = 50

5. The Constraint Method

Linearization of the model

As noted before, objective Z_2 is nonlinear. However, under some conditions on the cost factors b_{in}, if the multiobjective problem is solved using the constraint method, and if objective Z_2 is disaggregated by facilities, the problem can be linearized.

The constraint method consists in constraining $(m-1)$ of the m objectives of the problem to have "acceptable" values. The remaining objective is optimized, subject to the original constraints of the problem, plus the extra $(m-1)$ new constraints. The "acceptable" values are modified and the problem solved again, as many times as needed for obtaining adequate trade-off curves. In our case, the model has two objectives to be minimized. The first objective is left as such (the travel costs objective), while the nonlinear congestion costs objective is disaggregated by facilities and each component constrained to have a value of at most C_j. This value is modified in successive runs of the model, and a trade-off curve is drawn.

The model becomes:

$$\text{Min} \quad \sum_i \sum_j h_i c_{ij} x_{ij} \tag{6}$$

Subject to

$$\sum_i \sum_{n=0}^{K} h_i x_{ij} b_{in} P_n (\sum_i h_i x_{ij}) \le C_j \qquad \forall j \tag{11}$$

and subject to (2) to (5).

This version of the model has a linear objective and linear constraints, except for constraint (11). In order to linearize this constraint, we first note that b_{in} is always increasing (at least, non-decreasing) with n, because customers' waiting cost is increasing with the number of customers in the system (except for crowd – loving customers, case we are not including here). If, additionally, the waiting cost b_{in} does not depend on the origin of the customers, that is $b_{in} = b_n$ for all i, then the constraint can be written as:

$$\left(\sum_i h_i x_{ij}\right)\left(\sum_{n=0}^{K} b_n P_n (\sum_i h_i x_{ij})\right) \le C_j \qquad \forall j \tag{12}$$

The left hand side of constraint (12) strictly increases with $\sum_i h_i x_{ij}$ (or λ_j), which we prove next.

Suppose for the moment that $b_n = 1$ for all n. The second factor on the left hand side of constraint (12) becomes $\sum_{n=0}^{K} P_n(\sum_{u} h_i x_{ij})$. This expression is always equal to one, for all values of $\sum_i h_i x_{ij}$, because it is just the sum of all probabilities of numbers of customers in the queuing system. When $\sum_i h_i x_{ij}$ increases, the probabilities of more customers in the system (larger values of n) must increase. The probabilities of fewer customers in the system (smaller values of n) must then decrease in the same amount, to keep the total probability sum equal to one.

We now relax the requirement that $b_n = 1$, and recall that b_n is non-decreasing with n. Since higher order probabilities are multiplied by numbers b_n that are larger (or at least, equal), $\sum_{n=0}^{K} b_n P_n(\sum_i h_i x_{ij})$ is non-decreasing with $\sum_i h_i x_{ij}$. Since in constraint (12), this non-decreasing expression is multiplied by the strictly increasing expression $\left(\sum_i h_i x_{ij}\right)$, it is easy to see that the product is strictly increasing, which completes the proof.

As constraint (12) is strictly increasing with $\sum_i h_i x_{ij}$, there must exist a number $\hat{C}_j$, such that whenever:

$$\sum_i h_i x_{ij} \le \hat{C}_j \qquad \forall j, \tag{13}$$

constraint (12) necessarily holds true. Consequently, equation (13) is a linear expression that can be used instead of constraint (12) in the optimization formulation. For each value of C_j in (12), the value of $\hat{C}_j$ in the equivalent linear constraint (13) is found by computing the value of $\sum_i h_i x_{ij}$ for which equation (12) holds as equality.

The final, linear model for the case $b_{in} = b_n$ for all i, is:

$$\text{Min} \quad \sum_i \sum_j h_i c_{ij} x_{ij} \tag{6}$$

$$\text{s. t.} \quad \sum_j x_{ij} = 1 \quad \forall\, i \tag{2}$$

$$x_{ij} \le y_j \quad \forall\, i,j \tag{3}$$

$$\sum_j y_j = p \tag{4}$$

$$\sum_i h_i x_{ij} \le \hat{C}_j \qquad \forall j\,. \tag{13}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \tag{5}$$

$$y_j \in \{0,1\} \qquad \forall j$$

Solution, analysis and calibration of the number s of servers per facility

The usual constraint method uses Branch and Bound (or any feasible solution method for integer programming formulations) and the following steps:

1. Determine the values of the cost parameters (c_{ij} and b_n). These values can be found from polls or focus groups. Set values for C_j, s, K and μ_j. Note that potentially, s and K can be also facility-dependent.
2. Starting from the values of C_j and remaining parameters, compute the value of $\sum_i h_i x_{ij}$ for which equation (12) holds as equality. Set $\hat{C}_j$ to this value.
3. Solve the model.
4. Modify the value of C_j using an adequate step size and direction. Go to step 2 until a sufficiently detailed trade off curve can be drawn.

This procedure would provide the necessary trade-off curves. However, a deeper analysis can be performed, which is useful for the determination of some parameters while doing the design of the system. Furthermore, in the absence of reliable information on waiting cost functions, this analysis shows the effects of using different cost function shapes.

Note first that a particular value of the parameter $\hat{C}_j$ in equation (13) could be the result of more than one combination of values of the relevant parameters, namely service rate μ_j, maximum queue length K_j, number of servers in each center s_j, and queuing cost function $\{b_n\}$. Once $\hat{C}_j$ is computed and plugged in constraint (13), the particular combination of values that were used for its computation becomes unimportant from the point of view of the optimization problem, which becomes a capacitated p-median. Furthermore, the only relevant figures of the problem at this stage are the parameters $\hat{C}_j$ and p, as well as the demand rates h_i at each node and travel costs c_{ij} between demand nodes and potential center sites. Then, we can separately perform the following stages:

Stage 1. Using the known data (h_i, c_{ij}), solve the capacitated linear integer optimization problem, for all required values of p and using the full range of possible values of $\hat{C}_j$. As a result of this stage, for each p, there is a set of solutions. Each solution corresponds to a value of the travel cost objective, and the demand figures at each facility, for that value of the objective. In order to determine the full range, of relevant values of $\hat{C}_j$, needed to perform this stage, note that if we add both sides of constraint (13) over all possible location candidate nodes, we obtain

$$\sum_i h_i \sum_j x_{ij} \leq \sum_j \hat{C}_j \text{ , or}$$

$$\sum_i h_i \leq \sum_j \hat{C}_j \text{ .}$$

The left hand side of this expression is just the total demand in the network, *H*, which is known. Without loosing generality, and for the sake of simplicity, we assume $\hat{C}_j = C \;\; \forall j$. Since the number of facilities is limited to *p*, we have $H \leq pC$. Then, for a particular value of *p*, we can find the minimum value of *C = H/p*, solve the integer programming formulation, increase the value of *C*, solve again, and keep increasing this value in successive runs of the model. As the value of *C* increases, representing higher congestion costs, the value of the travel cost objective must either remain the same or decrease (improve), since we are relaxing a constraint of the problem (constraint 13). At some point, constraint (13) is not binding anymore, the solution corresponds to the unconstrained *p*-median solution, and we can stop.

In each solution, although the demand at each facility is bounded above by *C*, it does not necessarily reach this value, so each one of the located facilities can have its particular amount of demand.

Stage 2. For each solution, using the congestion cost function obtained from polls or focus groups, as well as the values of the parameters *K*, *s* and *μ*, compute the waiting cost at each facility. For different values of *K*, *s* and *μ*, different costs will be obtained, and consequently, by changing these values, a calibration of the design can be performed at this stage. Add these costs over all facilities, to obtain the total waiting cost for that solution.

Stage 3. Use the results of the Stages 1 and 2 to draw trade-off curves relating travel and congestion costs. If the cost are in monetary terms, find among all the solutions, the one with the least travel plus congestion cost. Note that the trade-off curves are not necessarily convex nor concave.

6. Computational experience on the constraint method

We solved the model using the constraint method and the linearization, as described above, on a 55-node network (Swain, 1974). The population at each demand node was used as a proxy for the demand rate at that node. The Euclidean distance between nodes was used as a proxy for the travel cost. We assume that the maximum queue length, *K*, at any facility or center, is 15 customers (no matter how many servers there are at that facility). Although we remark that the model accepts different values for each facility, for the example we assume that not only *C*, but the parameters μ_j and s_j are the same across the system, omitting the subscript *j*.

Recall that in practice, the queuing cost functions $\{b_n\}$ can be found by polling. In our example, in order to analyze what happens with different types of customers, we use two different possible congestion cost curves, that correspond to customers with different possible perceptions, represented by linear and quadratic cost functions, as shown in Table 3 and Figure 1:

	Cost functions $\{\boldsymbol{b_n}\}$	
Nr of customers in queue	Linear	Quadratic
0	1	1
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25
6	6	36
7	7	49
8	8	64
9	9	81
10	10	100
11	11	121
12	12	144
13	13	169
14	14	196
15	15	225

Table 3: Congestion Cost Functions

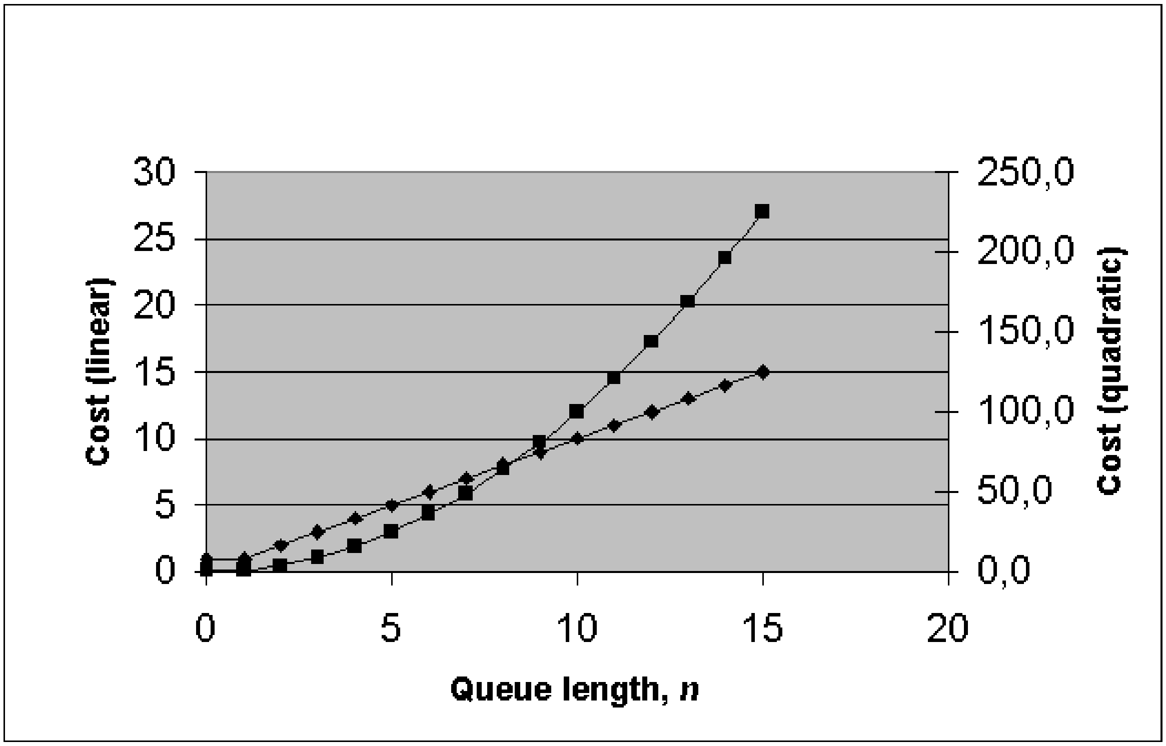

Figure 1: Different cost functions

Since the expected waiting time is proportional to the expected number of customers in queue (both mathematically and intuitively), those customers for which the most important issue is the time spent in queue, will most probably be represented by a linear cost function. The quadratic function, on the other hand, represents the perception of those customers who not only value the time spent in queue, but also dislike crowds or the feeling of standing on line, or in general, those who feel that standing in line is worse than spending time on any other activity. In this case, there is an extra cost that has to be added to the pure time-related cost.

In Stage 1, we solved the optimization model on a Pentium III, 1.6 GHz, 256 MB RAM computer, running MPL for Windows 4.12 and CPLEX 7.5, for $p = 2$, 3 and 6. The sets of solutions for $p = 2$, 3 and 6 are shown in Tables 4, 5 and 6.

C	Travel cost	Demand on facility 1	Demand on facility 2
3200	477.505	3200	3200
3300	476.300	3300	3100
3400	475.603	3390	3010
3500	475.554	3410	2990
3600	474.457	3580	2820
3700	473.594	3700	2700
3800	471.968	3790	2610
3900	471.916	3870	2530
4400	471.810	4360	2040
5000	467.427	5000	1400
5200	461.820	5200	1200
5230	461.477	5230	1170

Table 4: Set of solutions for $p = 2$

C	Travel cost	Demand on facility 1	Demand on facility 2	Demand on facility 3
2140	418.123	2140	2140	2120
2200	416.998	2190	2090	2120
2300	414.038	2290	2300	1810
2400	412.716	2400	2090	1910
2500	410.728	2500	2420	1480
2600	405.884	2600	2600	1200
2700	404.477	2690	2540	1170
2800	402.883	2640	2800	960
2900	401.432	2550	2890	960
3000	399.908	2450	2990	960
3100	398.912	3080	2150	1170
3300	398.416	3300	1930	1170
3400	394.788	3390	1810	1200
3500	393.474	3470	1760	1170
3700	390.616	3690	1540	1170
3900	388.283	3890	1200	1310
4040	387.024	4040	1170	1190

Table 5: Set of solutions for $p = 3$

For p = 2, the run times for the solutions were between 0.6 and 51 sec. The number of Branch and Bound nodes ranged between none and 55. For p = 3, the run times for the solutions were between 0.7 and 112 sec. The number of Branch and Bound nodes ranged between none and 255.

C	Travel cost	Demand on facility 1	Demand on facility 2	Demand on facility 3	Demand on facility 4	Demand on facility 5	Demand on facility 6
1090	297.84	1090	1090	1080	1090	980	1070
1100	296.88	1100	1090	1080	1100	960	1070
1200	289.127	1200	1200	1200	1110	1080	610
1300	280.11	1300	1300	1300	930	960	610
1400	275.523	1400	1320	1380	730	960	610
1500	273.385	1500	1260	1320	750	960	610
1600	272.165	1580	1220	1300	730	960	610
2500	268.925	2500	1380	930	510	470	610
2800	265.617	2800	1300	730	490	470	610

Table 6: Set of solutions for *p* = 6

For p = 6, the run times for the solutions were between 0.5 and 147 sec. The number of Branch and Bound nodes ranged between none and 774.

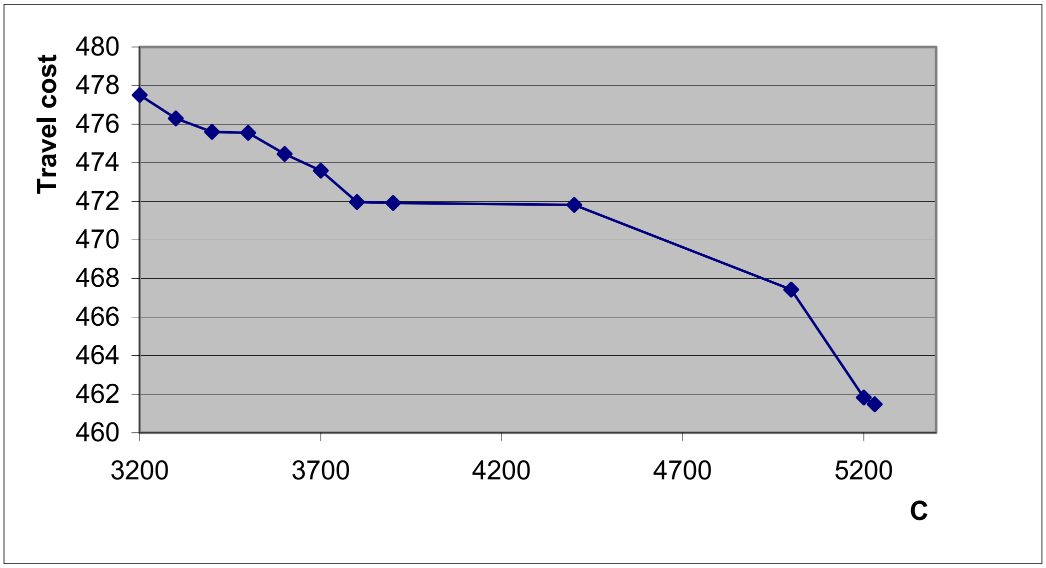

Figure 2: Trade off between travel costs and *C*, for *p* = 2.

For these cases, we computed the trade – off curves between the demand limit, *C*, and the travel cost. The curves are shown in Figures 2, 3 and 4. Note that these curves are neither convex nor concave, due to the integer nature of the problem.

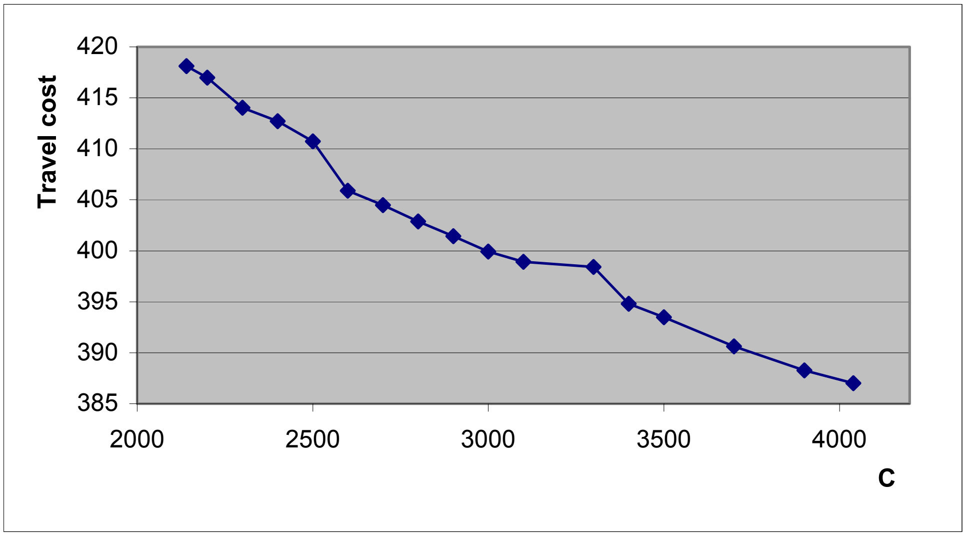

Figure 3: Trade off between travel costs and *C*, for *p* = 3.

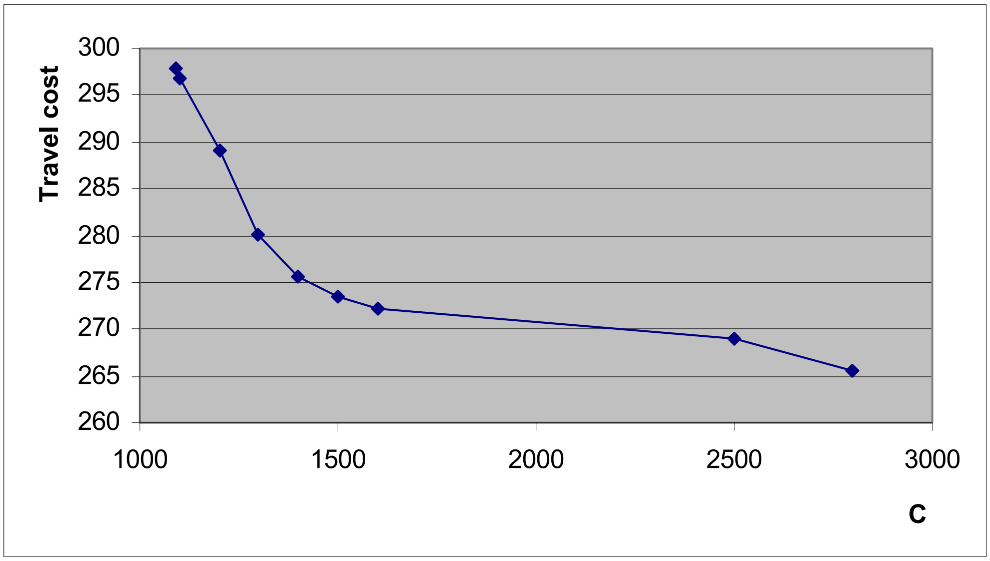

Figure 4: Trade off between travel costs and *C*, for *p* = 6.

C	Travel cost	Linear congestion costs		Quadratic congestion costs	
		facility 1	facility 2	facility 1	facility 2
3200	477.505	48.846	48.846	631.79	48.846
3300	476.300	51.426	46.161	673.24	46.161
3400	475.603	53.67	43.643	709.28	43.643
3500	475.554	54.159	43.07	717.13	43.07
3600	474.457	58.202	37.983	781.95	37.983
3700	473.594	60.95	34.167	825.85	34.167
3800	471.968	62.965	31.203	857.94	31.203
3900	471.916	64.726	28.521	885.93	28.521
4400	471.810	75.089	13.468	1049	107.2
5000	467.427	87.983	4.1444	1249.6	15.526
5200	461.820	91.93	2.9204	1310.5	8.5929
5230	461.477	92.519	2.771	1319.6	7.8715

Table 7: Costs for all solutions, $p = 2$, $\mu = 800$, $s = 3$.

C	Travel cost	Linear congestion costs			Quadratic congestion costs		
		facility 1	facility 2	facility 3	facility 1	facility 2	facility 3
2140	418.123	11.71	11.71	11.24	86.47	86.47	81.6
2200	416.998	12.93	10.57	11.24	99.53	74.57	81.6
2300	414.038	15.57	15.85	5.77	128.90	132.05	29.0
2400	412.716	18.72	10.57	7.18	165.40	74.57	41.4
2500	410.728	21.70	19.31	2.95	201.27	172.41	8.5
2600	405.884	24.70	24.70	1.85	238.55	238.55	3.3
2700	404.477	27.37	22.90	1.77	272.49	216.08	3.0
2800	402.883	25.89	30.54	1.31	253.63	313.62	1.7
2900	401.432	23.20	33.04	1.31	219.80	346.52	1.7
3000	399.908	20.20	35.70	1.31	183.09	382.01	1.7
3100	398.912	37.99	11.95	1.77	412.87	88.99	3.0
3300	398.416	43.22	7.50	1.77	484.05	44.40	3.0
3400	394.788	45.22	5.77	1.85	511.56	29.02	3.3
3500	393.474	46.96	5.18	1.77	535.32	24.16	3.0
3700	390.616	51.51	3.30	1.77	597.80	10.61	3.0
3900	388.283	55.44	1.85	2.20	651.65	3.27	4.7
4040	387.024	58.30	1.77	1.83	690.61	2.98	3.2

Table 8: Costs for all solutions, $p = 3$, $\mu = 800$, $s = 3$.

C	Travel cost	Quadratic congestion costs					
		Facility 1	Facility 2	Facility 3	Facility 4	Facility 5	Facility 6
1090	297.84	17.40	17.40	16.56	17.40	10.1	15.75
1100	296.88	18.30	17.40	16.56	18.30	9.2	15.75
1200	289.127	30.15	30.15	30.15	19.23	16.6	1.86
1300	280.11	48.99	48.99	48.99	7.94	9.2	1.86
1400	275.523	76.87	53.80	70.51	3.16	9.2	1.86
1500	273.385	114.41	40.47	53.80	3.45	9.2	1.86
1600	272.165	150.43	33.28	48.99	3.16	9.2	1.86
2500	268.925	562.31	70.51	7.94	1.21	1.0	1.86
2800	265.617	664.73	48.99	3.16	1.10	1.0	1.86

Table 9: Quadratic costs for all solutions, $p = 6$, $\mu = 800$, $s = 2$.

In the example network, $\sum_i h_i = H = 6{,}400$, so for two facilities the minimum possible value of C is 3,200. Below this value, the problem becomes infeasible. The p-median solution corresponds to $C = 5{,}230$.

In Stage 2, we computed the waiting cost of each solution, for linear and quadratic congestion cost curves. For $p = 6$, only the quadratic costs were computed.

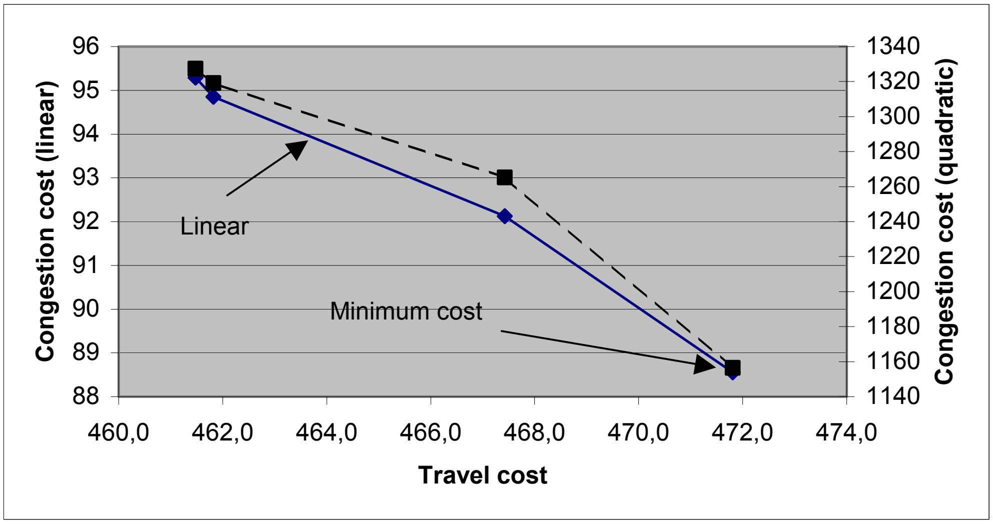

Figure 5: Trade off between travel and congestion costs, for $p = 2$.

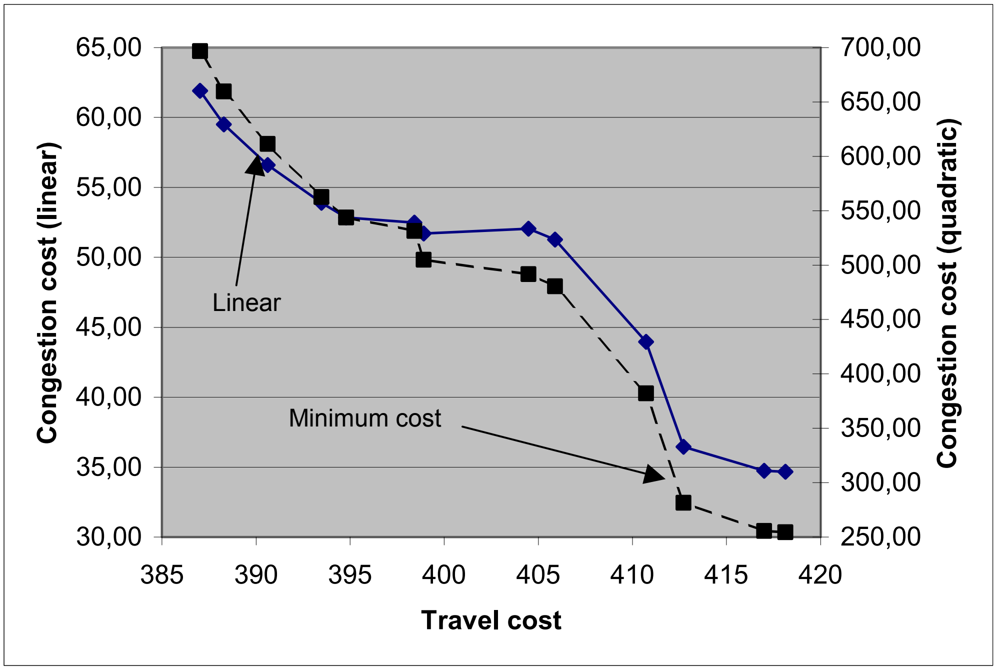

Figure 6: Trade off between travel and congestion costs, for *p* = 3.

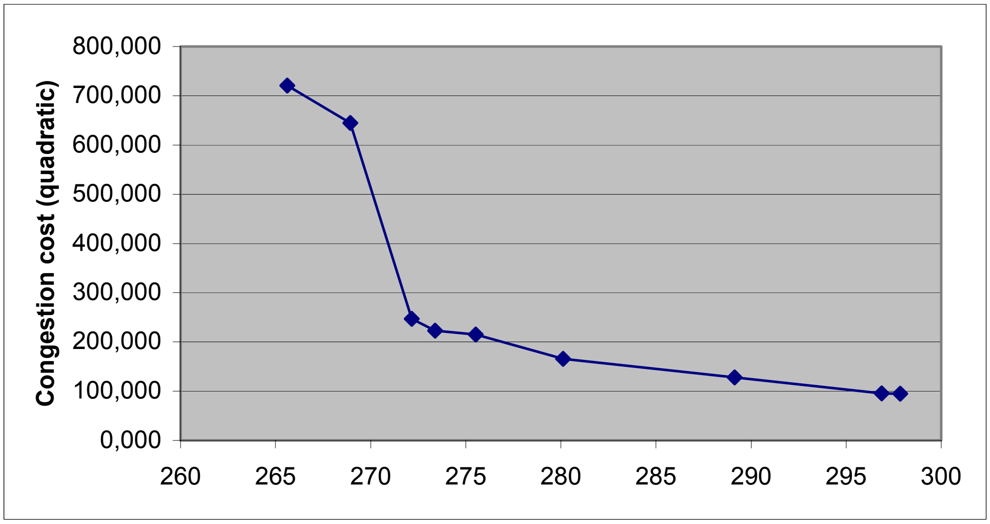

Figure 7: Trade off between travel and congestion costs, for *p* = 6.

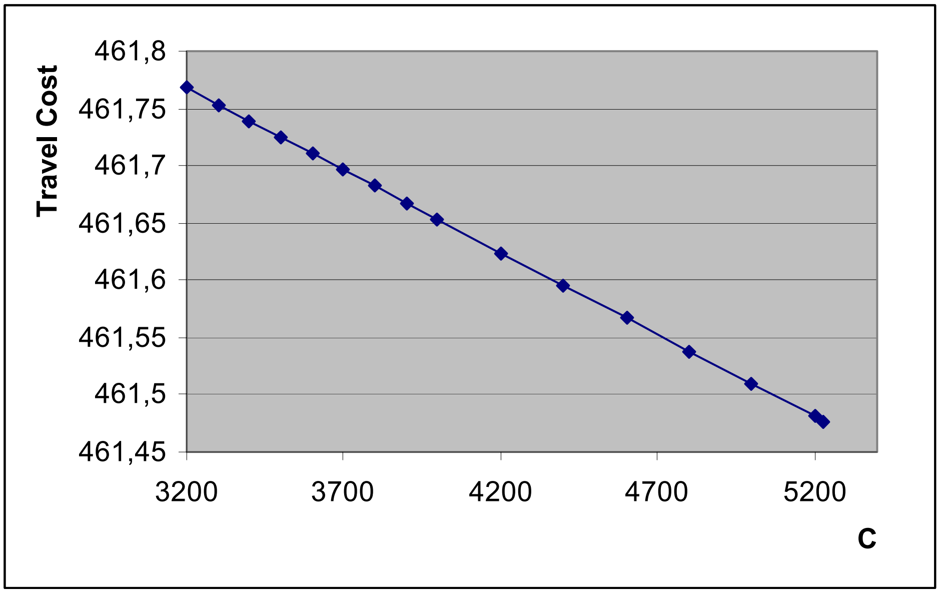

Figure 8: Congestion cost curve for $\mu = 800$, $s = 3$

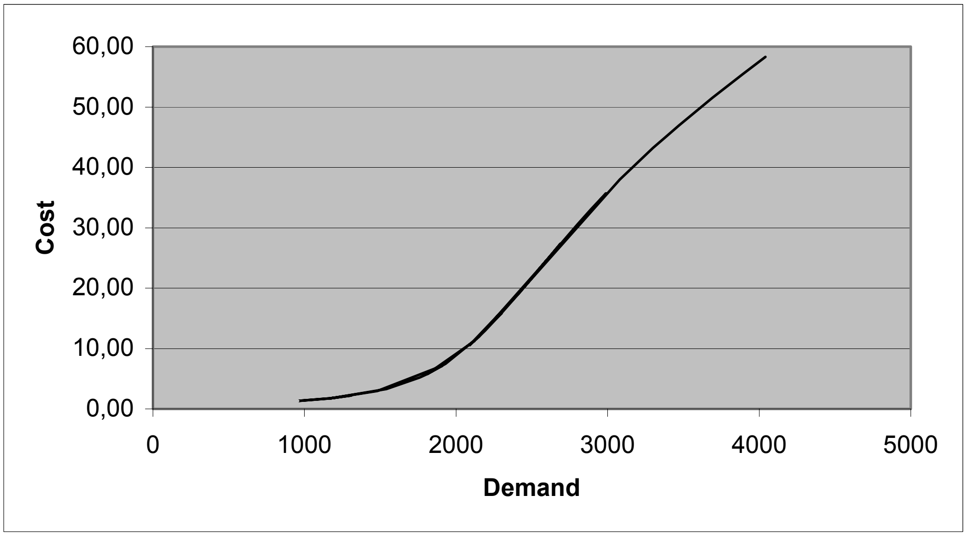

Figure 9: Travel cost trade-off

Note that the number of servers per center is different in the cost computation for $p = 6$. The maximum queue length, K, was set to 15 for all cases. In each case, the costs for the linear case and for the quadratic case were added, in order to compute the total congestion cost for that particular solution.

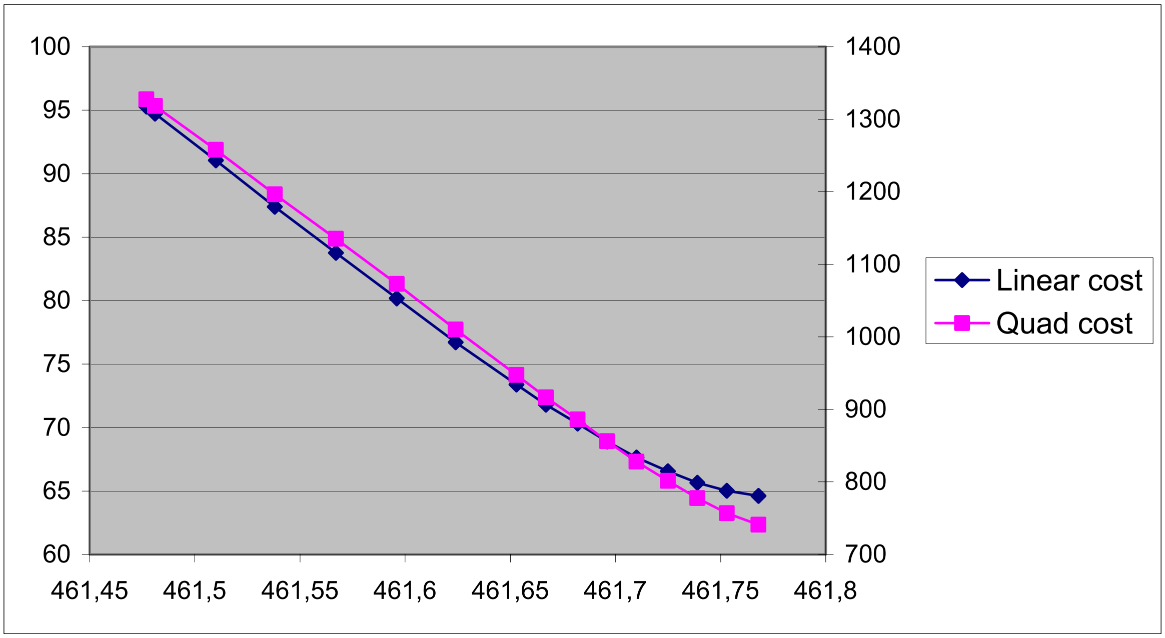

Figure 10: Trade off between travel and congestion costs, for $p = 2$, linear relaxation.

As a result of Stage 3 of the procedure, we show on Figures 5 to 7, the trade-off curves, in which only the non-dominated or Pareto optimal solutions are shown.

As an example of computation of the optimal solution when both costs are cast in monetary terms, in Figures 5 and 6, the solutions corresponding to the least total cost are marked with an arrow. Note that the shapes of all the trade-off curves are a result mainly of the integer nature of the variables, enhanced by the neither concave nor convex congestion costs. For example, part of the congestion cost curve for $\mu = 800$, $s = 3$ is shown in figure 8.

As an example of the effect of the integer nature of the variables on the shape of the trade-off curves, Figures 9 and 10 show the trade-off curves for the linear relaxation of the problem, with $p = 2$:

As expected, the shapes of the trade-off curves of the relaxed versions of the problem are closer to the usual shapes.

7. Conclusions

We propose a model that locates a fixed number of multiple-server service centers or public facilities, both susceptible of becoming congested. Consumers' perceived congestion and travel costs are minimized, considering that all the demand must be served. The travel cost is a general function of the origin and destination of the demand, while the congestion cost is a general function of the number of customers in queue at the facilities, which is proportional to the waiting time. Since the model is nonlinear, we investigate two different solution methods, useful under different circumstances. The first one is based on a Max-Min Ant System and Tabu Search, and finds a solution (not necessarily optimal) in a short time, particularly if the objectives can be easily expressed in the same units. The second procedure is somewhat slower, but allows solving the problem when the relative weights on the objectives are not known, and also allows a deeper analysis of different solutions, particularly if different numbers of servers can be located at different centers. Computational experience is presented on both solution procedures, using in each case such instances of the problem, as to allow a better evaluation of both methods.

APPENDIX 1: Metaheuristic for the multiobjective nonlinear formulation

Notation:

- LOCP is the set of center locations.
- Z_1 is the value of the travel cost objective and Z_2 is the value of the congestion cost objective
- W_1 and W_2 are the preset weights on the objectives, that is, the weighted objective Z is computed as $Z = W_1 Z_1 + W_2 Z_2$

Procedure:

1. Initialize the parameters of the Ant System and compute the initial pheromone trail

$$\tau_j = \frac{1}{\sum_i h_i c_{ij}}$$

for all nodes j of the network. Make $k = 0$.

2. Let $k = k + 1$ (iterations of the Ant System procedure)

First Step (FS): Construct a good initial solution

FS.1. Compute $p_j = \frac{\tau_j}{\sum_{l \in J} \tau_l}$, for all nodes j of the network. Make $i = 0$.

FS.2. Let $i = i + 1$

FS.3 Choose randomly a node to locate a center, using the probability function defined in step FS.1.

FS.4 If $i < p$, go to step FS.2.

FS.5 Compute both objectives and find the weighted objective Z

Second Step (SS): Local search phase; Tabu Search**.**

SS.1. Let $s = 0$ (number of times, the diversification criteria is applied).

SS.2. Let $t = 0$ (number of iterations of the TABU procedure).

SS.3. Set $Z^0(\text{LOCP}) = Z^{\text{BEST}}(\text{LOCP})$, the best solution found in MAX-MIN Ant system. Set LOCP_i^0, for $i = 1,.., p$ the optimal locations found in MAX-MIN Ant system.

SS.4. $\text{LOCP}_i^0 \rightarrow \text{LOCP}_{i+1}^0$ (for each located node).

SS.5. Consider all neighborhood nodes j of optimal location node i (i.e. $\text{LOCP}_{\text{ngh j}(\text{Locp}_i^0)}$).

Let $\text{LOCP}_{\text{ngh j}(\text{Locp}_i^0)} \rightarrow \text{LOCP}_{\text{ngh j+1}(\text{Locp}_i^0)}$. Exchange the facility from node LOCP_i^0 $\in$ LOCP to a neighborhood node $\text{LOCP}_{\text{ngh j}(\text{Locp}_i^0)} \notin$ LOCP.

Check the threshold constraint in this new solution:

- If this is satisfied, compute the objective function $Z(\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)})$.
- If this is not satisfied, set $Z(\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}) = 0$.

Do it for all neighborhoods nodes of node i.. Re-label the solution $\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}$ in decreasing order of $Z(\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)})$. When all neighborhoods are visited, go to SS.7

SS.6. If ($Z(\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}) > Z_{\mathrm{BEST}}(\mathrm{LOCP}_i^0)$) or $\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}$ is not tabu, then set $Z_{\mathrm{BEST}}(\mathrm{LOCP}_i^0) = Z(\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)})$, the outlet is located in $\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}$ and LOCP_i^0 is declared tabu until $t + \varepsilon$, where ε is a pre-fixed value, and go to step SS.5.

If all nodes visited are tabu and none improves the objective, then the model chooses the node with the lowest tabu tag ($t + \varepsilon$) and lift the tabu status of $\mathrm{LOCP}_{\mathrm{ngh}\ j(\mathrm{Locp}_i^0)}$). Then go to step SS.5.

SS.7. If t is less than a pre-fixed upper bound T; update the best solution found $Z_{\mathrm{BEST}}(\mathrm{LOCP})$, let $t \to t + 1$ and go to step SS.3.

8. If $s < s^{\mathrm{MAX}}$ (maximum number of times, the diversification criteria is applied). The model starts a new procedure with an initial solution equal to the NP least visited nodes. Let $s = s + 1$ and go to step SS.2.

Third Step (TS): Update pheromone trails, using the current solution.

TS.1. Compute the news τ_j for all nodes j of the network, using the optimal locations found in the previous steps. Check also the max – min limits for all τ_j (i.e. $\tau_{\min} \le \tau_j \le \tau_{\max}, \forall j$).

TS.2. If k < MAXITER, set $k \to k+1$ and go to step 2 of the initial procedure. Updating the Best Solution found $Z_{\mathrm{BEST}}(\mathrm{LOCP})$ in each MAX-MIN Ant system iteration.

Comments about the meta-heuristic:

- In step 6 of Tabu Search, tabu status can be canceled if this permits an improvement in the objective. This rule is called *the aspiration criterion*
- Step 8 of Tabu Search, states *the diversification criterion,* that allows a broader exploration of the solution space by starting from locations that have been less well explored.

Aknowledgements

This research has been possible thanks to research grants by FONDECYT (Project Nr 1020778, Chile) and by MCYT (Project BEC2000-1027, Spain)

References

Benati, S and Laporte, G. “Tabu Search algorithms for the (r | X_p)- Medianoid and (r | p)-Centroid Problems“. *Location Science*, 2(4), 1994, pp. 193-204.

Brandeau M., Chiu S., Kumar S., Grossman T., 1995, "Location with Market Externalities" In *Facility Location: A Survey of Applications and Methods*, Zvi Drezner (ed), 570 pp., Springer Verlag, New York NY.

Cohon J., 1978: *Multiobjective Programming and Planning*, Academic Press, Inc.

Colorni, A. Dorigo, M. and Maniezzo, V. “Distributed optimisation by Ant Colonies”, *Proceeding of ECAL91 – European Conference on Artificial Life:* Elsevier Publishing, Paris, France, 1991a, pp. 134-142.

Colorni, A. Dorigo, M. and Maniezzo, V. “The Ant System: Optimisation by a Colony of Cooperating Agents”, *IEEE Transactions on Systems, Man and Cybernetics – Part B,* 26(1), 1991b, pp. 29-41.

Dorigo, D. and DiCaro, G. “The ant colony optimization meta-heuristic”. In D.Corne, M.Dorigo and F.Glover (eds), *News Ideas in Optimisation,* 1999, McGraw-Hill.

Gendreau, M.; Hertz, Z. and Laporte, G. “A Tabu Search Heuristic for the Vehicle Routing Problem“. *Management Science* 40(10), 1994, pp.1276-1289.

Glover. F. “Tabu Search, part I“. *ORSA Journal of Computing,* 1, 1989, pp.190-206

Glover. F. “Tabu Search, part II“. *ORSA Journal of Computing,* 2:, 1990, pp. 4-32.

Hillier F S, and Lieberman G J, 1986, *Introduction to Operations Research*, Holden-Day, Inc., Oakland, CA.

Jamil M, Baveja A, Batta R, 1999, “The Stochastic Queue Center Problem”, *Computers and Operations Research,* **26**, 1423-1436.

Klincewicz, J.G. “Avoiding Local Optima in the p-Hub location problem using Tabu Search and GRASP“. *Annals of Operations Research,* 40, 1992, pp. 283-302

Lourenço, H. and Serra, D. "Meta-heuristics for the Generalized Assignment Problem". In *Mathware & Soft Computing, Special Issue on Ant Colony Optimization: Models and Applications*, forthcoming

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

Marianov V., Serra D., 2001, "Hierarchical Location-Allocation models for Congested Systems", *European Journal of Operational Research*, **135**(1), 196 – 209.

Marianov V., Serra D., ReVelle C., 1999, “Location of hubs in a competitive environment” - *European Journal of Operations Research*, **114**, 363 - 371.

Rolland E, Amiri A, Barkhi R, 1999: “Queuing Delay Guarantees in Bandwith Packing”, *Computers and Operations Research*, 26, 921 – 935.

Rolland, E.; Schilling, D.A. and Current, J.R. “An efficient Tabu Search procedure for the p-Median problem“. *European Journal of Operation Research* 96(2), 1997, pp. 329-342.

Stüzle, T. “An ant approach for the flow shop problem”. *In proceeding of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT’98),* 3, 1998, pp. 1560-1564.

Stüzle, T. “Local Search Algorithms for Combinatorial Problems – Analysis, Improvements, and New Applications”. PhD thesis, Department of Computer Science, Darmstadt University of Technology, Germany, 1998.

Stüzle, T. “MAX-MIN Ant System for the Quadratic Assignment Problem”. *Technical Report AIDA-97-4, 1997, FG Intellektik, TU Darmstadt, Germany.*

Stüzle, T. and Hoos, H. “Max-Min Ant System and Local Search for Combinatorial Optimisation”, in S.Voβ, S. Martello, I.H. Osman and C. Roucairol (eds.), *Meta-Heuristics: Trends in Local Search paradigms for Optimisation,* 1999, Kluwer Academic Publishers, pp. 313-329.