

Locating Emergency Services with Different Priorities: The Priority Queuing Covering Location Problem ¹

Francisco Silva^a, Daniel Serra^b

^a GREL, IET, Universitat Pompeu Fabra, Ramon Trias Fargas, 25-27, 08005 Barcelona Spain.
CEEApIA, Universidade dos Açores, Rua da Mae de Deus, 9502 Ponta Delgada, Portugal
email: francisco.silva@upf.edu

^b GREL, IET, Universitat Pompeu Fabra, Ramon Trias Fargas, 25-27, 08005 Barcelona Spain.
email: daniel.serra@upf.edu

Abstract

Previous covering models for emergency service consider all the calls to be of the same importance and impose the same waiting time constraints independently of the service's priority. This type of constraint is clearly inappropriate in many contexts. For example, in urban medical emergency services, calls that involve danger to human life deserve higher priority over calls for more routine incidents. A realistic model in such a context should allow prioritizing the calls for service.

In this paper a covering model which considers different priority levels is formulated and solved. The model inherits its formulation from previous research on Maximum Coverage Models and incorporates results from Queuing Theory, in particular Priority Queuing. The additional complexity incorporated in the model justifies the use of a heuristic procedure.

Keywords: Location, health services, queuing, heuristics.

JEL: C61, L80

¹ This research has been possible thanks to the grant SFRH/BD/2916/2000 from the Ministério da Ciência e da Tecnologia, Fundação para a Ciência e a Tecnologia of the Portuguese government, and grant BEC2000-1027 from the Ministerio de Ciencia y Tecnologia, of the Spanish government.

Introduction

Questions related to emergency services have been studied by researchers over the last 25 years. They refer to medical systems, police operations, firefighting systems, emergency repair systems, and others. Researchers agree that the outcome of service is in great part defined by the time that the customer waits.

Emergency services planners must solve the strategic problem of where to locate emergency service centers and the tactical problem of allocating demand to those centers. The performance of an emergency center may be judged by the number of persons in queue or by the length of time that a person must wait after he or she arrives at the center. These indicators are strongly correlated with the number of centers available and with their locations. Not all cases have the same dependence on time: rush jobs are taken ahead of other jobs, and important customers may be given precedence over others. This is clearly the case in hospital emergency rooms, where patients are roughly divided into three categories: critical cases, where prompt treatment is vital for survival; serious cases; and stable cases, where treatment can be delayed without adverse medical consequences.

Reducing waiting lists is a struggle for many healthcare organizations, especially given that the cost of the resources demands a high utilization rate. A lengthy patient wait in the healthcare industry shows more adverse consequences than in most other services. This generates stress and dissatisfaction, increases the cost of medical care, and can constitute a barrier to effective treatment. An optimized location for all facilities and equal allocation of patients to those facilities is a vital factor in improving time performance. This paper addresses emergency healthcare management, but the research can easily be expanded to other areas such as distribution centers or repair systems.

Marianov and Serra (1998) introduce the queuing maximal covering location allocation model, which locates p centers and allocates users to these centers in order to maximize the covered

population. Coverage is defined as (i) when patients are allocated to a center within a standard time or distance from home location, and (ii) when patients are served within time τ of arrival at the center, with a probability of at least α . This paper presents a natural expansion of this model that considers different time standards for different priorities of healthcare services.

Decisions that do not account for different priorities may lead to less effective locations. This paper considers different priority levels. The formulation of this model derives from previous research on maximum coverage models and incorporates results from queuing theory, in particular priority queuing. The additional complexity incorporated in the model justifies the use of a heuristic procedure. We develop an evaluation based on numerical examples.

1 Related Literature

Location models that incorporate queuing effects appear in the literature of the early 1980s. Berman, Larson, and Chiu (1985) present a work that is considered as the “beginning in a potentially fecund marriage between location and queuing theories.” They extend Hakimi’s (1964) one-median problem by embedding it in a general queuing context. The formulation explicitly includes dependence on service times, travel times, and queuing delays on the location of the service facility. Their work was in part motivated by the pioneer hypercube queuing model developed by Larson (1974).

Batta, Larson, and Odoni (1988) note that queuing disciplines frequently used in decision models (such as first-come first-served, last-come first-served, and service in random order) are clearly inappropriate in many contexts. They point to urban emergency services and police patrols as examples of when the risk of life or the violence of the crime will factor into the service order. A formulation for the single server queuing location model is provided (k priority queuing location) as well as some solution techniques that allow calls to be selected from an arbitrary number of priority classes. The optimal K-PQL model location usually differs from the

one obtained by grouping calls from all priorities into a single category and using the single queue length model.

Batta (1989) considers the problem of locating a single server on a network operating as an M/G/1 queue, in which queued calls are serviced by a class of queuing disciplines that depend solely on expected service time information. The model is analyzed as an M/G/1 non-preemptive priority queuing model, with location-dependent priorities. Motivated by the problem of locating fire trucks in a geographical area, which requires multiple trucks to be located within an acceptable distance standard to achieve coverage, Batta and Mannur (1990) examine the set covering problem and the maximal covering location problem in the context of multiple units being required by some demands.

For congested service systems, Brandeau and Chiu (1992) present the stochastic queue center location model, which seeks to minimize the maximum expected response time to any customer. Expected response time comprises expected waiting time until the server becomes free and expected travel time. A more recent work in the same line of research was developed by Jamil, Baveja, and Batta (1999). The stochastic queue center problem considers the objective of locating a single facility operating as an M/G/1 queue in steady state so as to minimize a weighted linear combination of the square of the average response time and the variance of the response time. Berman and Vasudeva (2000) also consider the problem of locating a general number of service units, which return to their home locations only if no calls are waiting for service.

Branas and Reville (2001) developed the trauma resource allocation model for ambulances and hospitals as a guide for healthcare planners. The model combines a mixed-integer linear program with a new heuristic and considers two resource, trauma centers and aeromedical depots, in a two-level hierarchy. The objective is to maximize coverage, which is defined as when at least one trauma center is sited within a ground standard time or when an aeromedical

depot/trauma center pair is sited in such a way that the sum of the flying time from the aeromedical depot to i and the flying time from i to the trauma center is within the same time standard.

Ball and Lin (1993) propose a reliability model for emergency service vehicle location. Based on the probability of system failure, they derive a 0-1 integer programming optimization model that is solved using a branch-and-bound procedure. The computational results show that the processing technique is highly effective. Mandell (1998) formulates a covering type model for two-tiered emergency medical services that maximizes the expected number of calls for service and takes server availability into account through a two-dimensional queuing model. Considering a redeployment problem for a fleet of ambulances, Gendreau, Laporte, and Semet (2001) propose a dynamic model and a parallel tabu search heuristic.

Harewood (2002) offers a multi-objective version of the maximum availability location problem in a real application by solving the problem of emergency ambulance deployment in Barbados. The first objective seeks to maximize the population covered within a given distance standard and with a given level of reliability, while the second objective chooses locations that minimize the cost of covering the population. Verter and Lapierre (2002) address the problem of locating preventive healthcare facilities. They assume that distance is a major determinant of participation and that people will go to the closest facility for preventive care. Golderg (2004) offers a review of the development and current state of operations research for deployment and planning analysis pertaining to emergency services and fire departments.

2 Formulation

The formulation of the model presented here closely follows the methodology proposed by Marianov and Reville (1994). The authors relax the assumption of dependence of server availability and model the behavior in each region as an M/M/s-loss queuing system obtaining a

probabilistic formulation of the location set covering problem. The reliability constraints are formally incorporated using queuing theory to model the arrival-departure process within the location model itself. The authors refer to fixed facilities, in contrast with earlier research that considers a mobile server. Marianov and Serra (1998) present several probabilistic, maximal covering, and location-allocation models with constrained waiting times for queue length. In this section, we propose a model that connects the queuing maximal covering location allocation model with priority queuing theory. We describe the results from priority queuing that will be included in the model and explain each of the equations in this new model.

2.1 Results from queuing theory

Priority queuing has been analyzed in many research works. In this paper, we follow Kleinrock's (1975) textbook notation. In a priority queuing system, we assume that an arriving customer belongs to priority class k ($k=1,2,\dots,K$). The smaller the priority index, the higher the priority of the class. Let us consider nonpreemptive priorities, i.e. when a customer in the process is not liable to be ejected from service and returned to the queue when a higher priority customer appears. Customers from priority k arrive in a Poisson stream at rate $\lambda^{[k]}$ for time unit. Each customer from this group has his or her service time S selected independently from the distribution $B_k(S)$ with mean $\bar{S}^{[k]}$. Let us also consider a Head of Line (HOL) discipline within each priority level. The average waiting time for priority k services is given by the following expression:

$$W^{[k]} = \frac{W_0}{(1-\sigma_k)(1-\sigma_{k-1})} \quad \text{if} \quad 1-\sigma_k > 0 \quad (1)$$

$$= +\infty \quad \text{otherwise}$$

where

$$\sigma_k = \sum_{i=1}^k \rho^{[i]} \quad \text{with } \sigma_0 \equiv 0$$

and

$$\rho^{[i]} = \frac{\lambda^{[i]}}{\mu^{[i]}} = S^{[i]} \lambda^{[i]}$$

$\mu^{[i]}$ is the service rate for priority class i .

The interpretation of ρ here is the fraction of time that the server is busy (as long as $\rho < 1$).

W_0 corresponds to the average delay that the tagged customer experiences due to another customer found in service and can be computed using the following formula:

$$W_0 = \sum_{i=1}^R \frac{\lambda^{[i]} \overline{(S^{[i]})^2}}{2}$$

where $\overline{(S^{[i]})^2}$ is the second moment of service time distribution.

2.2 Problem definition

The structure of the problem comprises a discrete space in which customers, representing demand, are positioned and in which facilities are to be positioned by a decision maker. Customers are assumed to be located at the nodes of the network (demand nodes). We assume that the service call process for each node i is a Poisson process with rate λ_i . The problem will locate a given number of facilities providing the service. We assume that a discrete set of potential facility locations eligible to provide coverage to the demand nodes has been identified.

For the purposes of this model, utilization factors are defined by the product of the average service time and the arrival rate as defined by the summation of the call rates of all demand points allocated to a center at j

$$\rho_j^{[k]} = \overline{S_j^{[k]}} \lambda_j^{[k]} = \overline{S_j^{[k]}} \sum_i f_i^{[k]} X_{ij}^{[k]}$$

where $X_{ij}^{[k]}$ is a zero one variable defining allocation of demand point i to a center at j for priority k services. $f_i^{[k]}$ is the call rate for services of priority class k at demand point i . For example, $f_i^{[k]} = 0.08$ indicates that at demand point i , the number of calls per unit of time for services of priority k equals 8 percent of the total population on average.

2.3 Priority queuing covering location problem

The priority queuing covering location problem (PQCLP) assumes static assignments of customers to service facilities. This is typical in the case of fixed server locations, where customers travel to facilities to obtain service (see Berman and Krass, 2002). We offer two versions of the PQCLP: the first assumes a directed choice environment in which a central authority dictates the assignment of a customer (patient) to a center, while the second assumes a user choice environment in which the assignment is made by each individual customer. In our case, they will always go to the closest center.

2.3.1 Directed choice environment

This version of the PQCLP will define separate allocations for the different priorities that may or may not coincide, i.e., a demand point may be allocated to a center at j for one priority and to center $k \neq j$ for other priorities. The objective is to maximize the population covered in all priorities. Different time constraints are imposed for different priorities. The model's formulation is as follows:

Max

$$Z = \sum_k \sum_i \sum_j a_i X_{ij}^{[k]}$$

s.t.

$$X_{ij}^{[k]} \leq Y_j \quad \forall i \in I, \forall j \in N_i, \forall k \quad (2)$$

$$\sum_{j \in N_i} X_{ij}^{[k]} \leq 1 \quad \forall i \in I, \forall k \quad (3)$$

$$\sum_j Y_j = p \quad (4)$$

$$W_j^{[k]} \leq \tau^{[k]} \quad \forall j, \forall k \quad (5)$$

$$X_{ij}^{[k]}; Y_j \in \{0,1\} \quad \forall i \in I, \forall j \in N_i \quad (6)$$

where $X_{ij}^{[k]} = 1$ if demand point i is allocated to a center at j for priority k 's urgencies and zero in all other cases. $Y_j = 1$ if a center is located at j and zero in all other cases. $W_j^{[k]}$ is the average waiting time for priority class k at center j (see Equation 1). Parameter $\tau^{[k]}$ represents the limits imposed on the waiting times for priority k ; p is the number of centers to be sited; a_i is the population at demand point i ; I is the set of all demand points; and N_i is the set of centers located at a distance smaller or equal to d from demand point i .

Constraint (2) states that if population i is allocated to a center at j for priority k , then there must be a center located at j ; (3) forces each demand point to be allocated to only one center; (4) defines the number of centers to be located; and (5) ensures that the average waiting time is less than the given standard. Additionally, j must be within set N_i so that a demand point requires a center located within distance d in order to be covered.

2.3.2 User choice environment

The above model can be converted to a user choice model by assuming that customers will always choose the closest center and by adding the following group of constraints to enforce this assumption:

$$X_{ij}^{[k]} \geq Y_j - \sum_{l \in C_{ij}} Y_l \quad C_{ij} = \{l | d_{il} < d_{ij}\} \quad \forall i \in I, \forall j \in N_i \quad (7)$$

Originally introduced by Rojeski and ReVelle (1970) in the context of the budget constrained median problem, this equation establishes that if j is an open center and no closer center is open, then demand i must be assigned to j . If j is open but a closer center is also open, then this relation does not constrain assignment in any way (for a more detailed discussion of closest assignment constraints, see Gerrard & Church, 1996).

Heuristic Solution Procedure

A greedy randomized adaptive search procedure (GRASP) is a metaheuristic that has been applied to a wide range of operations research and industrial optimization problems. These include problems of scheduling, routing, logic, partitioning, location, layout, graph theory, assignment, manufacturing, transportation, telecommunications, automatic drawing, electrical power systems, and VLSI design (see Resende, 1998). Festa and Resende (2001) present an extensive annotated bibliography of GRASP literature. The following notations are used in this section:

j index for potential locations

i index for demand nodes

k index for priority level

D_j list of potential facility location ordered in accordance with total population

S solution set

\bar{S} complement of the solution set

C set of candidate sites

p number of facilities to locate

n number of demand nodes

inc_k_j incoming call rate at potential facility location j

inc_j total incoming call rate at potential facility location j

ρ_k_j utilization factor at potential facility location j for priority level k

D_{ij} list of demand points within the standard distance from potential facility location j

R total priority levels

w_{k_j} waiting time for priority level k at potential facility location j

f_{k_i} frequency of demand for priority k services at demand node i

GammaSet set containing all possible values for γ

prob_γ probability of choosing a specific value for γ

$ut(\gamma)$ utility value of a specific γ value

GRASP is an interactive process with a feasible solution constructed at each independent iteration. One GRASP iteration consists of two phases: construction phase and local search phase. Figure 1 describes a pseudo code for the GRASP algorithm.

INSERT FIGURE 1 HERE

The procedure for the construction phase, which will return an initial solution at each of the iterations, is denoted as *Greedy_Randomized_Construction(Seed, γ)* and is a function of the seed in the random numbers generation and of a parameter gamma defining which solutions are included in the restricted candidate list (RCL), which contains the best greedy solutions.

The development of this procedure appears in Figure 2.

INSERT FIGURE 2 HERE

The algorithm starts by sorting the candidate nodes in accordance with their populations. In our example, every demand point is considered as a potential facility location. Another possibility would be to consider only a subset of the demand points in the list D_j . Beginning with the first candidate on the list, we allocate the closest demand points to each node until the coverage limit is reached. This limit can be reached by the utilization coefficient or by the limit imposed on the waiting time at each priority level.

The total demand assigned to each of the potential locations j is notated as the incoming call rate. Since there are different waiting time limits corresponding to the different priority levels,

there are also different incoming call rates. The total incoming call rate is defined as the sum of the incoming call rates for all priorities. The incoming call rates are the greedy function of the algorithm and can be defined as the weight of as yet uncovered demand points that would become covered if center location j were to be chosen.

We include in the RCL the candidate nodes with total incoming call rates greater or equal than γ percent of the incoming call rate corresponding to the potential facility location with the higher value. In the GRASP algorithm, parameter γ is defined a priori (e.g. for a γ value equal to 0.8, we include all of the potential locations with total incoming call rate greater than 80 percent of the highest incoming call rate). At each of the iterations, we randomly choose the facility locations within the candidate sites with the highest incoming call rates. Note that in a pure greedy heuristic, like the one suggested by Marianov and Serra (1998), the choice would always be to locate a center in the node with the highest sum of incoming call rates ($\gamma=1$).

In the local search phase, we de-allocate the demands that were allocated to each center and move the center to all possible unused potential facility sites, repeating steps 9 through 20 of the greedy randomized construction procedure. If some locations give a better objective, the center should remain at that location; otherwise, the facility should return to its initial location (see Figure 3). This procedure continues until no further improvements can be made.

INSERT FIGURE 3 HERE

To avoid fixing parameter γ in an arbitrary manner, we have implemented a reactive procedure in which γ is updated at each one of the iterations. Reactive GRASP, proposed by Prais and Ribeiro (2000), is a procedure in which the parameter is self-adjusted according to the quality of the solutions previously found. Instead of fixing the value of γ to determine which elements will be placed in the RCL, R-GRASP randomly selects this parameter value from a discrete set $\{\gamma_1, \dots, \gamma_m\}$. The probability distribution used in the γ selection will be updated after the

execution of each block of iterations considering the quality of the solutions obtained by each of the γ_i (the utility of the γ_i). Figure 4 describes the R-GRASP algorithm.

INSERT FIGURE 4 HERE

In a user choice environment, the heuristic algorithm must be adapted in the construction and local search phases to enforce closest assignment. In this new version, a demand point will always be assigned to the closest center location, which may lead to infeasible solutions such that the limits imposed for the average waiting times will not be verified. The proposed algorithm penalizes the objective when an infeasible solution is obtained. When obtaining a feasible solution, this set of locations is considered as a potential center location. Otherwise, this set is considered only as an initial solution and not as a potential location penalizing the objective with a large negative value M . This corresponds to the following objective procedure in figure 5.

INSERT FIGURE 5 HERE

In the local search phase, a demand point is always allocated to the closest potential center location, and the feasibility of waiting time limits is checked. If the solution is not feasible, then the objective is penalized with M . When new locations result in a better objective, the center is placed at that location; otherwise, it remains at the initial location. This procedure is repeated until no further improvements can be made.

4 Computational Experience

4.1 Evaluation of Heuristic

4.1.1 Deviation from the optimal solution

To observe the difference between the results from the heuristic and the optimal solution, a simple experiment is implemented that consists of randomly generating problem instances and

comparing results from the heuristic with those from complete enumeration. The coordinates of the demand points and the potential facility sites are randomly generated from a uniform distribution between 0 and 6. The populations for each demand point are generated from a uniform distribution between 60 and 800. In all examples, only two priorities are considered. Call rates for first and second priorities are 0.5 percent and 1 percent, respectively, of the populations. A waiting time limit of 5 minutes is imposed for first priority and 20 minutes for second priority.

The objective values obtained using the first heuristic are compared with those obtained through a complete enumeration procedure in which allocations of demand points to potential facility sites are defined by repeating steps 1-5 of the heuristic algorithm described in Section 4.1. The objective values obtained from the second heuristic are compared with those from a complete enumeration procedure that always allocates a demand point to its closest center. For Heuristics 1, we consider $\gamma=1$ (i.e., the pure greedy algorithm). For Heuristic 2, the pure greedy algorithm does not behave well, and it was necessary to implement GRASP.

In these examples the number of blocks (*nblocks*) was 4, each with a number of iterations (*niter*) equal to 50. The average deviations and maximum deviations are computed for the 50 instances generated. The main results for the different combinations of the number of demand points and the number of centers appear in Table 1.

INSERT TABLE 1 HERE

4.1.2 Computing time

To evaluate the heuristic's behavior, we measure processing times for larger problem instances. Table 2 shows the average computing times for simulated instances with 50 and 100 demand points (in a Pentium(r) III processor with 128 MB of RAM).

INSERT TABLE 2 HERE

4.2 Illustrative example

For simplicity, we consider only two priorities: critical and stable. Additionally, we assume that the first and second moments of service time are the same for both priorities and are independent from the demand's location; we also assume that customers must travel to the center to obtain service and that service time will not vary among locations. In the computational experience, the 30-node problem instance (Marianov & Serra, 1998) is used. We authors assume that the servers are physicians, that each demand point is also a potential center location, and that the distances are Euclidean.

Figure 1 represents each demand point as a square which area signals the relative population at that demand point. Demand points are numbered from 1 to 30 in decreasing order of population. For our purposes, demand is separated into the daily call rate for first priority cases (0.005 times the population) and the daily call rate for second priority cases (0.01 times the population). The instances are adequate for the problem in analysis. The more populated demand points are grouped in the center; this eliminates the location of all centers in the nodes with higher frequencies. The instances seem to be quite representative of many urban areas, where most of the population is concentrated in the center with a variety of smaller populated areas around the center. The results of imposing a waiting time limit of 5.5 minutes on the first priority and 20 minutes on the second priority appear in Table 3.

INSERT TABLE 3 HERE

Table 4 shows the results of considering one priority with an average time limit of 12.75 minutes. The average service time in this example is 10 minutes. The values on Table 4 were computed using the heuristic procedure proposed by Marianov and Serra's (1998). Although 9

centers would cover the entire population, considering only one priority, Table 3 shows that 1.1 percent of the population would not receive first priority service within the time limit. This percentage increases drastically with a decrease in the number of centers: with only 5 centers, 14.8 percent of the population would not receive first priority service within the time standard. Location decisions will vary considerably depending on whether or not different priorities are taken into account. For example, in the 10 center case, only 6 coincide in both scenarios.

INSERT TABLE 4 HERE

Figure 6 illustrates the 9 center case; nodes with a center are represented by non-dashed squares. Centers are not located in the demand points with the highest population but are strategically distributed.

INSERT FIGURE 6 HERE

Considering separate allocation variables for different priorities, it is possible for a demand point to be allocated to more than one center. At the same time, some centers may receive many calls for some priorities and no calls for others. A center that offers first priority service will be located at j if (and only if) $\exists i \in N_j : X_{ij}^{[1]} \neq 0$. A center that offers second priority service will be located at j if (and only if) $\exists i \in N_j : X_{ij}^{[2]} \neq 0$ where $N_j = \{i | d_{ij} \leq d\}$.

Figure 7 presents the results of the 9 center case. Only 6 centers (represented by the double circle) are needed to cover both priorities if 3 centers (represented by the single circle) that offer service only to first priority cases are located. Note that these centers are located closer to the demand points with the highest population and are needed in order to avoid congestion among first priority services.

INSERT FIGURE 7 HERE

In this model, the need for service dictates the number of centers and the type of services offered by each center. This can lead to important resource savings. Table 5 illustrates the results obtained by applying Heuristic 2 to the previous example. The waiting time limits are 5.5 minutes for first priority cases and 20 minutes for second priority cases. Demand points are always assigned to the closest facility independently from the priority level; therefore, the percentage covered on first and second priorities coincide. In this new situation, 8 centers are needed to cover the entire population.

INSERT TABLE 5 HERE

5 Conclusions and Future Research

When waits are common, utilization rates are high, and resources are expensive, considering more than one priority helps to create a more efficient geographic distribution of service centers. Some centers accumulate more customers of a low priority; emergency centers, when strategically located, can avoid congestion of higher priority services. The PQCLP links population needs with resource allocation. These results should be applied to a directed choice environment in which there is some control over the assignment of demand points to the located centers.

The proposed heuristic procedures provide results that are coherent with the rationality of the model. An extensive evaluation of these procedures shows that the solutions are quite close to those obtained by complete enumeration. The heuristic procedure shows important advantages in terms of computational speed. It allows us to find solutions for problems that may otherwise require a prohibitive amount of computer time. Priority queuing is a useful theory for many types of services and should be extended to other location-allocation models.

References.

- Batta, Rajan (1989). A queuing-location model with expected service time dependent queuing disciplines. *European Journal of Operational Research* **39**: 192-205.
- Batta, Rajan and Narasimha R. Mannur. (1990) Covering-Location Models for Emergency Situations That Require Multiple Response Units. *Management Science*. **36**: 16-23.
- Batta, Rajan, Richard C. Larson and Amadeo R. Odoni, (1988). A Single-Server Priority Queuing-Location Model. *Networks*. **8**: 87-103.
- Ball, Michael O. and Feng L. Lin. (1993). A Reliability Model Applied to Emergency Service Vehicle Location *Operations Research*. **41**: 18-36.
- Berman, O. and Dmitry Krass (2002). Facility Location Problems with Stochastic Demands and Congestion. *Facility Location: applications and theory*. Springer Verlag.
- Berman, Oded, Richard C. Larson and Samuel S. Chiu. (1985). Optimal Server Location on a Network Operating as an M/G/1 Queue. *Operations Research*. **12**: 746-771.
- Berman, O. and S. Vasudeva. (2000). Approximating Performance Measures for Public Services”, *working paper*, Joseph L. Rotman School of Management, University of Toronto.
- Branas, Charles C. and Charles S. Revelle. (2001). An iterative switching heuristic to locate hospitals and helicopters. *Socio-Economic Planning Sciences*. **35**: 11-30.
- Brandeau, M. L. and S.S. Chiu. (1992). A center location problem with congestion. *Annals of operations research*. **40**: 17-32.
- Festa, P and M.G.C. Resende. (2001). “GRASP an annotated bibliography.” In *Essays and Surveys on Metaheuristics*, edited by P. Hansen, and C.C. Ribeiro. Norwell, MA: Kluwer Academic Publishers.
- Gendreau, M., G. Laporte and F. Semet (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocations. *Parallel Computing*. **27**: 1641-1653.
- Gerrard, Ross A. and Richard L. Church. (1996). Closest Assignment Constraints and Location Models: Properties and Structure. *Location Science*. **4**, 4.
- Goldberg, J.B. (2004). “Operations Research Models for the Deployment of Emergency Services Vehicles”. *EMS Management Journal*. **1**, 1.
- Jamil, M. , Alok Baveja and Rajan Batta. (1999). The stochastic queue center problem. *Computers & Operations Research*. **26**: 1423-1436.
- Kleinrock, L. (1975). Queuing Systems. Volume I: Theory. John Wiley and Sons.

- Hakimi, S. (1964). Optimal Location of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*. **12**: 450-459.
- Larson, R.C. (1974). A hypercube queuing model for facility location and redistricting in urban services, *Comp. Oper. Res.* **1**: 67-95.
- Mandell, M.B. (1998). Covering models for two tied emergency medical systems. *Location Science*. **6**: 355-368.
- Marianov, Vladimir and Daniel Serra. (1998). Probabilistic, Maximal Covering Location-Allocation Models for Congested Systems. *Journal of Regional Science*. **38**: 401-424.
- Marianov, Vladimir and Charles ReVelle. (1994). The Queuing Probabilistic Location Set Covering Problem and Some Extensions. *Socio-Economic Planning Sciences*. **28**: 167-178.
- Prais and Ribeiro. (2000). Reactive GRASP: an application to a matrix Decomposition Problem in TDMA traffic assignment. *INFORMS Journal on Computing*. **12**: 164-176 (2000).
- Resende, Mauricio G.C.(1998). Computing Approximate Solutions of the Maximum Covering Problem with GRASP. *Journal of Heuristics*. **4**: 161-177.
- Rojeski, G. and C. ReVelle (1970). Central facilities location under an investment constraint *Geographical Analysis*, **2**: 343-360.
- Verter, Vedat and Sophie D. Lapierre, (2002). Location of Preventive Health Care Facilities. *Annals of Operations Research*. **110**: 123-132.

Table 1

Computational Results- Deviations.

		Average Deviation (% of complete enumeration objective)		Maximum Deviation (% of complete enumeration objective)	
		Heuristic 1	Heuristic 2	Heuristic 1	Heuristic 2
20 nodes	2 centers	0.07%	0.29%	2.48%	11%
	3 centers	0.41%	0.30%	5.04%	12%
	4 centers	0.46%	1.64%	9.95%	21%
30 nodes	2 centers	0.03%	0.84%	1.71%	10%
	3 centers	0.39%	0.57%	4%	14%

Table 2

Computational results- Average computing times.

Number of Demand points	Number of Centers	Average Computing Time	
		Heuristic 1	Heuristic 2
50	5	15.44	67.07
	10	24.17	321.81
100	5	143.13	580.95
	10	207.07	1706.92

Table 3

Heuristic's results for one server per center and two priorities.

Number of Centers	Objective	Locations	% pop covered (1 st priority)	% pop covered (2 nd priority)
10	10 940	1;6;8;11;14;18 20;22;24;29	100%	100%
9	10 880	1;6;8;11;18;20; 22;24;29	98.9%	100%
8	10 810	1;6;8;11;20;22; 24;29	92.1%	96.9%
7	10 650	1;6;8;11;20;22; 29	90.7%	95.4%
6	10 350	1;6;8;20;22;29	86.3%	95.4%
5	10 160	1;2;20;22;29	85.2%	95.4%

Table 4

Heuristic's results for one server per center and one priority.

Number of Centers	Objective	Locations	% pop covered
10	5 470	1;3;5;6;7;14 17;18;20;24	100%
9	5 470	3;5;6;7;14;17; 18;20;24	100%
8	5 400	3;5;6;7;14;17; 20;24	98.7%
7	5 320	3;5;6;7;14;17; 20	97.3%
6	5 140	3;5;6;7;17;20	94.0%
5	4 950	5;6;7;17;20	90.0%

Table 5

Heuristic's results for one server per center and two priorities with closest assignments.

Number of Centers	Objective	% pop covered (1 st priority)	% pop covered (2 nd priority)
10	10 940	100%	100%
9	10 940	100%	100%
8	10 940	100%	100%
7	10 800	98.7%	98.7%
6	10 780	98.5%	98.5%
5	6 200	56.6%	56.6%


```
procedure GRASP (Max_iterations, Seed)
1   For k=1 to Max_iterations do
2        $S \leftarrow \text{Greedy\_Randomized\_Construction}(Seed, \gamma)$ ;
3        $S \leftarrow \text{Local\_Search}(S)$ ;
4        $\text{Update\_Solution}(S, \text{Best\_Solution})$ ;
5   enddo
6   end GRASP
```

Figure 1: GRASP Pseudo Code

procedure *Greedy Randomized Construction* (Seed, γ)

```

{sort candidate sites by decreasing order of population}
1    $D_j \leftarrow \text{Sort\_Candidate\_Sites}(\text{population});$ 
{initialize solution set}
2    $S := \{\};$ 
3    $\bar{S} := C;$ 
{while solution is not a complete solution}
4   while  $|S| \neq p$  do
    {loop over all candidate sites not in the solution list}
5     For  $j=1$  to  $|\bar{S}|$  do
      {initialize parameters}
6        $inc\_k\_j := 0; inc\_j := 0; W\_k\_j := 0; \rho\_k\_j := 0;$ 
      {restrict demand points list to the standard covering distance to site j}
7        $D_{ij} \leftarrow \{i \in D, d_{ij} \leq d\}$ 
      {sort demand points by increasing distance to site j}
8        $D_{ij} \leftarrow \text{Sort\_Demand\_Points}(distance);$ 
      {loop over priorities}
9       For  $k=1$  to  $R$  do
        {loop over demand points in set  $D_{ij}$ }
10        For  $i=1$  to  $|D_{ij}|$  do
          {sum frequencies at each demand point if waiting time limit is not reached}
11          If  $(W\_k\_j < \tau^{(k)} \text{ and } \rho\_k\_j < 1)$  do
12             $inc\_k\_j := inc\_k\_j + f\_k\_i;$ 
13            actualize  $w\_k\_j;$ 
14            actualize  $\rho\_k\_j;$ 
15          endif
16        enddo
        {compute incoming call rate at the potential center location}
17         $inc\_j := inc\_j + inc\_k\_j;$ 
18      enddo
      {construct the restricted candidate list}
19       $c^{max} := \max\{inc\_j\};$ 
20       $RCL \leftarrow \{j \in \bar{S}, inc\_j \geq \gamma c^{max}\};$ 
      {select randomly one site from the RCL}
21       $j^* \leftarrow \text{Random\_Select}(RCL);$ 
22       $S := S \cup \{j^*\};$ 
23       $\bar{S} := \bar{S} \setminus \{j^*\};$ 
      {take the demand points allocated to  $j^*$  out of the demand point's list}
24      For  $i=1$  to  $|D_{ij^*}|$  do
25         $D := D \setminus \{i \in D_{ij^*}\};$ 
26      enddo
27    enddo
28  end Greedy Randomized Construction

```

Figure 2: Construction Phase Pseudo Code

```

procedure Local_Search (Solution, Best_Solution)

1   obj_best := obj(S);
   {loop over sites in the solution}
2   for all  $j \in S$  do
3        $S := S \setminus \{j\}$ ;
   {loop over sites not in the solution}
4       for all  $j^* \in \overline{S}$  do
5           evaluate obj( $S \cup \{j^*\}$ );
6           if obj_best < obj( $S \cup \{j^*\}$ ) do
7                $S := S \cup \{j^*\}$ ;
8               obj_best := obj( $S \cup \{j^*\}$ );
9           else
10               $S := S \cup \{j\}$ ;
11          endif
12      enddo
13  enddo
14  end Local_Search

```

Figure 3: Local Search Phase Pseudo Code

```

procedure reactive_GRASP
    {initialize gammas' probabilities}
1   For all  $\gamma \in \text{gammaSet}$  do
2        $prob\_ \gamma \leftarrow \frac{1}{|\text{gammaSet}|}, \forall \gamma \in \text{gammaSet};$ 
3   enddo
4   For  $block=1$  to  $nblocks$  do
    {implement GRASP algorithm}
5       For  $iter=1$  to  $niter$  do
6            $\gamma \leftarrow \text{Random\_Select}(\text{gammaSet}, \text{prob});$ 
7            $S \leftarrow \text{Greedy\_Randomized\_Construction}(\gamma);$ 
8            $S \leftarrow \text{Local\_Search}(S);$ 
9            $\text{Update\_Solution}(S, \text{Best\_Solution});$ 
10      enddo
    {actualize gammas' probabilities after each block of iterations}
11       $prob\_ \gamma \leftarrow \frac{ut(\gamma)}{\sum_{all\gamma} ut(\gamma)}, \forall \gamma \in \text{gammaSet};$ 
12   enddo
13   end reactive GRASP

```

Figure 4: reactive GRASP Pseudo Code

```

procedure evaluate_objective (S)
1   Allocate each demand point to its closest center location;
2   Evaluate  $W_{k_j}$  and  $\rho_{k_j}$ ;
3   obj(S):=0;
4   If ( $W_{k_j} < \tau^{[k]}$  and  $\rho_{k_j} < 1$ ) do
5       For k=1 to R do
6           For j=1 to p do
7               For i=1 to n do
8                   If (i is allocated to j) do
9                       obj(S):=obj(S) + fk_i;
10                  endif;
11              enddo;
12          enddo;
13      enddo;
14  else
15      obj(S):=M;
16  end evaluate_objective;

```

Figure 5: Objective Evaluation Pseudo Code

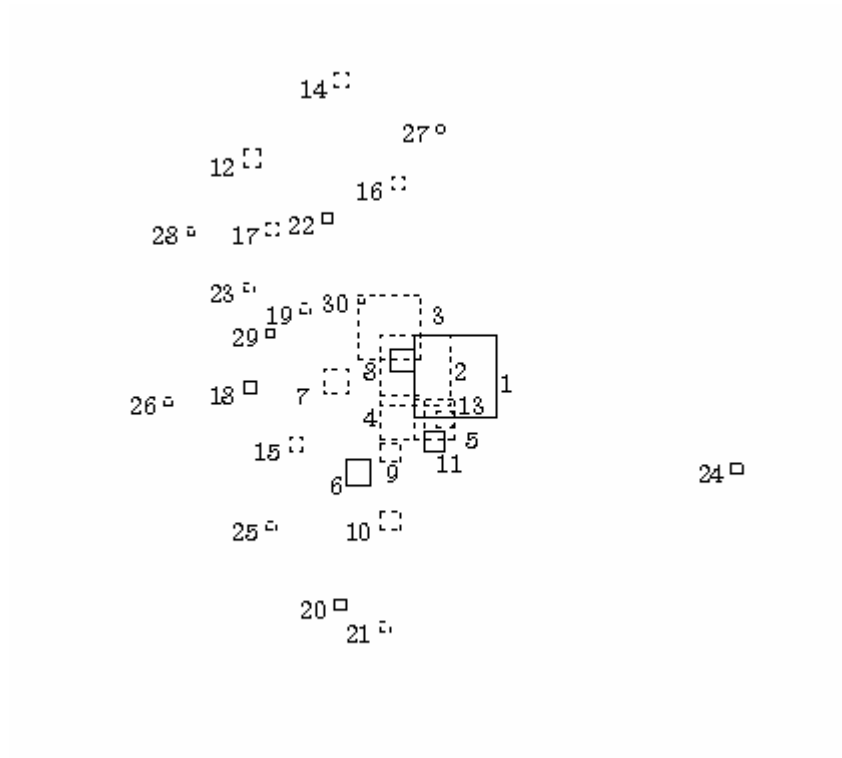


Figure 6: Locating the 9 centers- a)



Figure 7: Locating the 9 centers- b)