Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands

Phone: +31-53-4893400
Fax: +31-53-4893114

Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1696

Locating repairshops in a stochastic environment

J.C.W. van Ommeren, A.F. Bumb and
A.V. Sleptchenko[1]

December, 2003

ISSN 0169-2690

[1]EURANDOM and Department of Technology and Management, Eindhoven University of Technology

Locating repairshops in a stochastic environment

J.C.W. van Ommeren, A.F. Bumb and A.V. Sleptchenko*

January 5, 2004

Abstract

In this paper we consider a repair shop location problem with uncertainties in demand. New local repair shops have to be opened at a number of locations. At these local repair shops, customers arrive with broken, but repairable, items. Customers go to the nearest open repair shop. Since they want to leave as soon as possible, a (small) inventory of working items is kept at the repair shops. A customer immediately receives a working item from stock, provided that the stock is not empty. If a stockout occurs, the customer has to wait for a working item. The broken items are repaired in the shop and then put in stock. Sometimes, however, a broken item cannot be fixed at the local repair shop, and it has to be sent to a central repair shop. At the central repair shop the same policy with inventory and repair is used.

The problem that we focus on, is not only finding locations for the local repair shops, but also minimizing the stock levels at the shops, such that the fraction of customers that can leave the local shops without waiting (the so called fill rate), is above a prespecified level. We assume that the central repair shop is already opened, but that the repair capacity still has to be set. The local repair shops can be opened at a number of locations, which may have different repair capacities.

The goal is to minimize the total cost, that is the total cost for keeping the local shops operational, for the transport of items and for the inventory. For this minimizing problem, a local search heuristic with respect to the open locations, repair capacities and inventory levels is presented.

Keywords stochastic facility location, repairable items
AMS classification 90B06, 60K30

1 Introduction

In this paper we investigate the following problem in spare part management. In order to improve its service to the customers, a company decides to open some local repairshops close to them. There are several locations where the local repairshops can be placed. At a repairshop several servers can be installed and spare parts can be kept in inventory in order to insure a high service level. A repairshop can either repair a broken item and add it to the existing inventory or can send it for repair to a central repair facility. The position of the central repairshop is known in advance, but the number of servers that will be installed there it is not. When deciding if at a certain location a repairshop will be opened, the company looks at the following costs: the cost of opening the facility, the cost of installing the necessary number of servers at that facility, the distance from the

*EURANDOM and Department of Technology and Management, Eindhoven University of Technology, The Netherlands

customers (each customer should be assigned to the closest open facility), the cost of the necessary inventory and the transportation cost to the central repair facility. The company prefers a solution that insures, at a minimal cost, a high quality of service, given by a small probability that a customer has to wait.

The model presented in this paper is related to the area of facility location and the area of inventory control in multi-echelon models for repairable systems. Although both problems have received much attention separately, not much has been done on adressing them together.

In a facility location problem, having some information on the demand and on the possible location of facilities, one has to decide where to open facilities such that certain objectives are realized (*e.g.*, minimization of costs, maximization of the population covered, minimization of response time). The literature on facility location problems is very vast. For a survey on models and methods see the books edited by Drezner [14], Mirchandani and Francis [20] and the review done by Hesse Owen and Daskin [15].

In the recent years, the issue of uncertainty of demand and transportation was addressed in several papers. Many of them concern models for emergency systems, in which a server travels to the site of the emergency, as opposed to systems in which servers are fixed at certain locations ([5, 6, 8, 17]). In the case of more mobile servers, the algorithms developed generally use as a sub-algorithm the single-server model ([9, 10]). In [18], Marianov and Serra analyse the issue of locating servers at fixed locations when the number of requests for service follow some probabilistic distribution. Their goal was to maximize the population covered under the constraint that the probability of a long response time or the probability of long queues are small. In [19] they extend their analysis to the situation in which the number of facilities and servers needed to cover all the population is minimized. In [26], Wang, Batta and Rump propose a heuristic for finding the optimal location of facilities in order to optimise the traveling cost of the customers and their waiting costs. In their model there is an upperbound on the number of facilities allowed to be opened and on the allowable expected waiting time at a facility.

For literature on spare parts managenent, we refer to Sherbrooke [24], Muckstadt [21], Avsar and Zijm [3] and Sleptchenko [25]. In these papers the focus is on multi-echelon inventory systems, (in a multi-echelon system, inventory is stored at different locations). The papers by Sherbrooke and by Muckstadt, assume that the repair capacity is infinite, so that all items are repaired simultaneously. They present algorithms for optimizing inventory levels at the different locations, the so called (MOD)METRIC models. In the papers by Avsar and Zijm and by Sleptchenko, the repair capacity is finite, so sometimes items have to wait for repair. By assuming that the repair times are exponentially distributed, Avsar and Zijm can model the system as a product form network. Sleptchenko uses an approximation based on the first two moments of the repair times. They analyse the behaviour of these systems, given a certain maximal inventory level, and use these analytic results to find the optimal inventory levels.

The paper is structured as follows. In Section 2 we describe the problem in more detail and propose a stochastic model for it. We model the repairshops as M/M/k queues and consider the transportation times deterministic. The quality of service will be given by imposing a small probability that a customer has to wait for service. Since it is very difficult to find this probability analytically, we will approximate it by using the method described by Avzar and Zijm in [3]. In Section 3 we propose a local search heuristic for finding a solution. In Section 4 we present some computational results illustrating the behaviour of the proposed procedure. We present our conclusions in Section 5.

2 The model

Next we will describe the problem of locating repairshops in a stochastic environment in more detail. There are a set of customers that require service (repair of a broken item), a set of location where local repairshops may be opened and an already opened central repairshop. We assume that the customers are grouped in clusters, depending on their geographical location and that the moments at which clusters request service form a Poisson process. Each cluster is assigned for service to the nearest open local repairshop.

At each local repairshop a stock of items is kept in order to replace the broken items brought by customers. The broken items that can be repaired locally are put back in stock and are ready to use, while the others are sent for reparation to the central repair shop. Here the same policy is used as in the local repair shops. At the moment a broken item arrives at the central facility, an item from the central stock is sent to the stock at the local repair shop. The broken item is repaired and put in the central stock. We assume that items do not have to wait for transportation to and from the central repairshop. At all repair shops, both local and central, several servers can be installed. Arriving requests which cannot be served immmediately, are put in a queue (backordered). At the local repair shops, these so called backorders, are of course unwanted. The probability that a customer does not have to wait for service is called the *fill rate*.

A scheme of a repairshop is displayed in Figure 1. The arrows on the left and right originate from, respectively point to clusters of customers. There are a number of exponential servers and three buffers. The buffers connected to the servers, contain items and the other buffer represents the waiting line of customers.

A similar scheme for the central repairshop is presented in Figure 2. The left and the right arrows originate from, respectively point to transportation nodes.

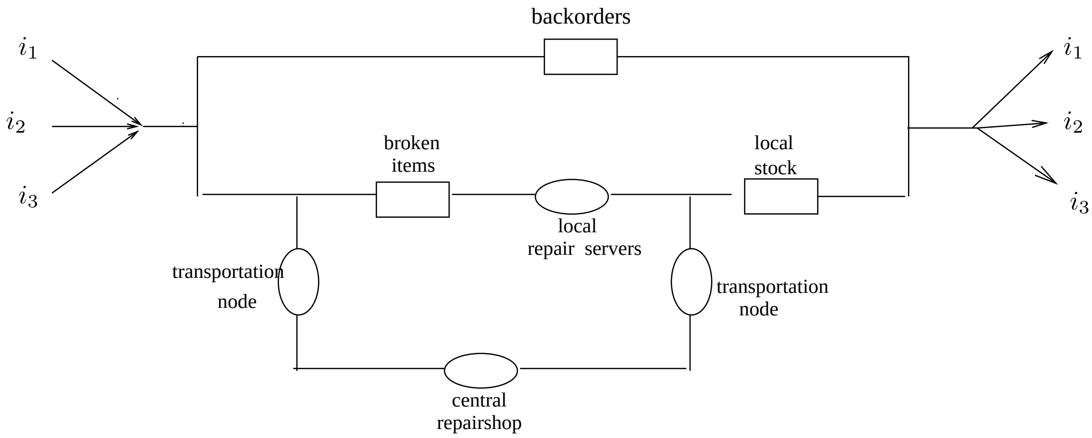

Figure 1: Local repairshop

One has to decide which facilities to open, how many repair servers to install and what the base stock levels should be, in order to insure a prespecified fill rate at the lowest expected cost. The costs that we consider are related to the stock levels, the opening of local facilities, the installation of repair servers, transportation from customers to local repair shops, transportation from the local repair shops to the central facility and vice

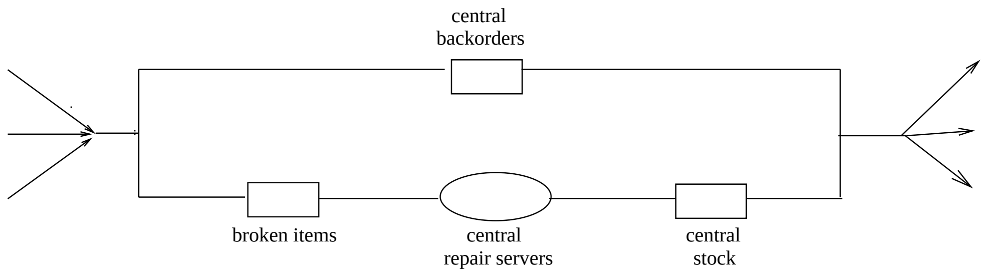

Figure 2: Central repairshop

versa. All the transportation costs are considered proportional to the distances.

To model this situation, we denote by :

- $D = \{1, ..., N\}$: set of clusters of customers (demand points);
- $F = \{1, ..., M\}$: set of locations where local repair shops can be opened;
- d_{ij} : distance between the cluster i of customers and location j;
- d_j^L : distance from location j to the central repair shop;
- f_j : cost of opening a repairshop at location j;
- S_j : cost of a stock unit at location j;
- S_C : cost of stock unit at the central location;
- s_C : cost of installing a server at the central facility;
- s_j : cost of installing a server at location j;
- w_L : the unit cost for the internal transportation from the local repair shops to the central repair shop;
- w_C : the external transportation cost of customers;
- λ_i : the rate at which requests for repair are generated at cluster i;
- λ_C : the overall arrival rate, i.e., $\lambda_C = \sum_{i \in D} \lambda_i$;
- μ_j : service rate at local repairshop j.
- μ_C : service rate at the central repairshop.
- α : the prescribed minimal value of the fill rate;
- χ : the probability that a broken item cannot be repaired to a local repair shop and is sent for repair to the central one.

For simplicity of the presentation, assume that $d_{ij} \neq d_{ik}$ for $j \neq k, j, k \in F$ and $i \in D$.

Let y_j $(j \in F)$ be variables indicating whether a repairshop at location j is open and x_{ij} $(i \in D,\ j \in F)$ variables indicating whether cluster i is assigned for service to facility j. Let V_j be the base stock level at a local repairshop $j, j \in F$ and let V_C be the stock level at the central repairshop. The number of servers installed at a location $j \in F$ will be denoted by k_j and the number of servers installed at the central repairshop by k_C. Clearly, the value of the vector x is completely determined by the vector y, namely

$$x_{ij} = \begin{cases} 1, & \text{if } y_j = 1 \text{ and } d_{ij} \leq d_{ik} \text{ for all } k \text{ such that } y_k = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Condition (1) can be rewritten as

$$\sum_{k \in F} d_{ik} x_{ik} \leq (d_{ij} - \Delta) y_j + \Delta,$$

where Δ is a large number, i.e., $\Delta = \max\{d_{ij} : i \in D, j \in F\}$.

We will model an open repairshop at location j as a queue with k_j exponential servers. At every repairshop, the arriving requests form a Poisson process with arrival rate $\sum_{i\in D} \lambda_i x_{ij}$. Both the input to the servers and the items sent from location j to the central location are filtered Poisson processes with rate $(1-\chi)\sum_{i\in D} \lambda_i x_{ij}$ and $\chi \sum_{i\in D} \lambda_i x_{ij}$, respectively. In order to satisfy the stability requirements, we impose that at every open repairshop the arrival rate should be less than the service rate. In other words, for each $j \in F$,

$$(1-\chi)\sum_{i\in D} \lambda_i x_{ij} \le \mu_j k_j y_j.$$

The total arrival rate at the central repairshop is $\chi \sum_{i\in D} \lambda_i = \lambda_C$, which we impose to be smaller then $k_C \mu_C$.
The total expected cost (the cost of opening facilities, the inventory cost and the expected transportation costs) can be then calculated by

$$\sum_{j\in F}(f_j + k_j s_j)y_j + S_C V_C + s_C k_C + \sum_{j\in F} S_j V_j y_j + 2\sum_{j\in F}\sum_{i\in D}(w_C d_{ij} + w_L d_j^L \chi)\lambda_i x_{ij}.$$

We arrive at the following mathematical programming formulation:

$$\begin{aligned}
\min \quad & \sum_{j\in F}(f_j + s_j k_j)y_j + S_C V_C + s_C k_C + \sum_{j\in F} S_j V_j y_j \\
& \qquad + 2\sum_{j\in F}\sum_{i\in D}(w_C d_{ij} + w_L d_j^L \chi)\lambda_i x_{ij} \\
s.t. \quad & \sum_{j\in F} x_{ij} = 1, \text{ for each } i \in D & (2)\\
& x_{ij} \le y_j, \text{ for each } i \in D \text{and } j \in D & (3)\\
& \sum_{k\in F} d_{ik} x_{ik} \le (d_{ij} - \Delta) y_j + \Delta, \text{ for each } i \in D, j \in F & (4)\\
& \sum_{i\in D} \lambda_i x_{ij} \le \mu k_j y_j, \text{ for each } j \in F \\
& \lambda_C \le k_C \mu_C \\
& Fillrate(V_j, k_j, \sum_{i\in D} \lambda_i x_{ij}, V_C, k_C) \ge \alpha y_j, \text{ for each } j \in F & (5)\\
& y_j \in \{0,1\}, \text{ for each } j \in F \\
& k_j, k_C, V_j, V \in Z, \text{ for each } j \in F
\end{aligned}$$

Constraints (2) and (3) insure that each cluster of customers is assigned to an open local repairshop, constraint (4) insures that each cluster of customers is assigned to the closest open repairshop and constraint (5) insures the required quality of service at an open repair shop.

2.1 An approximation of the fillrate

In this subsection we derive an approximation to the fillrate, that can be easily described analytically. All the stochastic variables we introduce are actually time dependent, but, since we are interested in the steady state behaviour of the system, we will omit this dependence. We introduce the following notations.

- N_C : number of items that are either being processed or waiting to be processed at the central location;
- N_j : the number of broken items that are either waiting to be processed or being processed at location $j \in F$;
- T_j : total number of items that is on transport from repairshop $j \in F$ to the central repairshop and vice versa;
- B_C : number of backorders at the central repairshop.
- B_{Cj} : number of backorders at the central repairshop originating at location $j \in F$. Note that $B_C = \sum\limits_{j \in F} B_{Cj}$.

The fillrate at location j, *i.e.,* the probability that a customer does not have to wait for service at a location j is given by

$$Fillrate(V_j, k_j, \textstyle\sum_{i \in D} \lambda_i x_{ij}, V_C, k_C) = Pr(N_j + T_j + B_{Cj} < V_j).$$

If we knew the distribution of $((N_j, T_j, B_{Cj}), j = 1, ..., N)$, we could easily calculate the fillrate. However, finding this distribution turns out to be a difficult task, since, due to the deterministic transportation times, the vector process is not even Markovian. If the transportation times were exponentially distributed, the process would indeed be a Markov process, but would still be intractable due to the large state space. As an alternative, we find a product form approximation for the distribution and analyse the random variables individually.

The simplification we apply, is described in Avsar and Zijm [3]. They replace the central facility, with its base stock policy, by a special state dependent server, which works with speed $min\{V_C + n, k_C\}\mu_C$, whenever there are $n > 0$ items at the central facility. If an arriving item finds the server free, the item is served at infinite speed with probability $q = \frac{\mathrm{P}(N_C < V_C)}{\mathrm{P}(N_C \leq V_C)}$ and therefore leaves the system immediately, otherwise it enters the system. In this modified model, the steady state distribution has a product form. The probability q is actually the probability that an item arriving to the central repairshop finds no other items waiting to be fullfiled while nevertheless the central stock is depleted.

In the following, suppose that the central facility is replaced by the special server as proposed in Avsar and Zijm [3]. For the new system, the fillrates can be calculated as follows.

Let j be a location where a local repairshop was opened. Using the product form of the solution, the fillrate at location j is

$$\mathrm{P}(N_j + T_j + B_{Cj} < V_j) = \sum_{r=0}^{V_j - 1} \sum_{s=0}^{r} \sum_{q=0}^{r-s} \mathrm{P}(N_j = s)\mathrm{P}(T_j = q)\mathrm{P}(B_{Cj} = r - s - q).$$

The random variable N_j is just the number of customers at an $M/M/k_j$ queue with arrival rate λ_j and mean service time $\frac{1}{\mu}$.

Due to the product form approximation, we can assume that at the transportation nodes, items arrive according to a Poisson proces with rate $\chi\lambda_j$. Since it is an ample server node, the number of used devices is Poisson distributed with expectation $\chi\lambda_j d_j^L$. Transportation being both to and from the central facility, the number of travelling items is Poisson distributed with expectation $\tau_j = 2\chi\lambda_j d_j^L$ and therefore

$$\mathrm{P}(T_j = k) = \frac{\tau_j^k}{k!} e^{-\tau_j}, \qquad k = 0, 1, \ldots.$$

Note that we again have used the product form approximation to conclude that transportation to and from the central facility location are independent processes. Finally, the steady state distribution of B_{Cj}, the number of backorders at the central repairshop originating at location j, can be calculated as follows:

$$Pr(B_{Cj} = i) = \sum_{m=n}^{\infty} Pr(B_{Cj} = i | B_C = n) Pr(B_C = n).$$

From the viewpoint of j, two types of items arrive at the central server, one originating at the local repairshop j and the another originating at the rest of the local repairshops. Denote the probability that an arrival at the special server is an item from j by p_j. Clearly,

$$p_j = \chi\lambda_j / \chi\lambda_C = \lambda_j / \lambda_C.$$

Moreover, due to the independence of the Poisson arrivals from different local repairshops, the probability that an item backordered at the central repairshop is coming from repairshop j is p_j. Hence,

$$Pr(B_{Cj} = i | B_C = n) = \begin{pmatrix} n \\ i \end{pmatrix} p_j^i (1 - p_j)^{n-i}.$$

The steady state distribution of the number of backordered items at the central repairshop is given by

$$Pr(B_C = n) = \begin{cases} Pr(N_C \leq V_C), & n = 0 \\ Pr(N_C = n + V_C), & n > 0. \end{cases}$$

Since the arrival process at the central location does not depend on the way customers are assigned to local repairshops, the number of items that have to be repaired at the central location is just the number of customers at a simple $M/M/k_C$ system with arrival rate λ_C and mean service time $\frac{1}{\mu_C}$.

Note that this direct calculation of the fillrate, though straightforward, is not very efficient for implementations, due to the large number of operations involved. A more efficient implementation will be presented in Section 4.

Next we will present a heuristic for finding a solution to the mathematical program presented, in which the fill rate is approximated as described in Section 2.1.

3 A local search heuristic

The model presented in the previous section is a variant of the capacitated facility location problem with additional constraints regarding the fillrate and the stability conditions at the central facility location. The capacitated facility location problem is considered very difficult (see ReVelle [23] for a discussion on the implication of the capacity constraints). Many methods have been proposed for tackling this problem, such as Lagrangian relaxation (*e.g.* [7, 11, 12]), polyhedral approach [1], branch and bound ([2, 13]), local search [16].

For deciding which facilities to open we will use a local search heuristic based on the procedure developed by Kuehn and Hamburger in [16].

The local search procedure starts with all facilities closed. In every step the operation with the largest cost improvement among the following three operations is performed: opening a new facility, closing an already opened facility or swapping (opening a closed facility and closing an opened one) facilities. The procedure stops when no improvement is possible.

When calculating the cost associated with a set of open facilities F_o, we take into account the following quantities: the costs of opening the facilities in F_o, the transportation costs, the inventory and server costs at the facilities in F_o and at the central facility. The necessary number of servers and the inventory (stock) at facilities in F_o and at the central facility is decided as follows.

Let min_cost be the minimum total cost associated with a set of open facilities that was analysed so far. Denote by $MC(F_o)$ the total cost associated with F_o when $V_C = \infty$ and $S_C = 0$. Clearly, $MC(F_o)$ is a lowerbound of the real cost associated with F_o. If $min\_cost < MC(F_o)$, we stop analysing F_o, since it cannot have minimal total cost. If $min\_cost > MC(F_o)$, we can interpret the quantity $min\_cost - MC(F_o)$ as being the available budget for F_o. Clearly, the available budget for F_o gives us the maximum number of servers , i.e., $\lfloor \frac{min\_cost - MC(F_o)}{s_C} \rfloor$ and the maximum stock at the central facility, i.e., $\lfloor \frac{min\_cost - MC(F_o)}{S_C} \rfloor$. The minimal number of servers at the central facility is given by $k_C = \lceil \frac{\lambda_C}{\mu} \rceil$ and the minimal stock is 0. For all the combinations of stock and servers at the central facility we calculate at each facility in F_o the cheapest combination of stock and servers for which the fillrate is above the prescribed value. If the total cost decreases below min_cost, we replace min_cost by the total cost of F_o. Note that, as in the case of the central facility, we can calculate an available budget for each facility in F_o. This budget will give an indication on the maximum stock level and the maximal number of servers that may be installed.

Next we will describe the algorithm in more detail, starting with the local search procedure and continuing with the procedures for calculating the costs related to the stock level and number of servers at the central facility ($Cost\_set\_facilities$), respectively at an open local facility (we will call it $Cost\_local\_facility$).

Local Search Procedure

$F_o = \emptyset$;
For each cluster of customers $i \in D$ find the closest facility and add it to F_o;
Let $min\_cost := \infty$
Until no cost improvement is possible, do
  Add a facility to F_o, Delete a facility from F_o,
  or Switch a facility from F_o with a facility in $F \setminus F_o$ such that
  $min\_cost := \min_{\{j \in F_o, l \in F \setminus F_o\}} \{Cost\_set\_facilities(F_o \cup \{j\}),$
  $Cost\_set\_facilities(F_o \setminus \{l\}), Cost\_set\_facilities((F_o \cup \{j\}) \setminus \{l\}\}$
  is attained
Return min_cost and the solution that has the total cost equal to it.

The following procedure optimizes the number of servers and the inventory level at the central repair shop.

$Cost\_set\_facilities(F_o)$

For each $i \in D$ assign i to the closest facility $j_i \in F_o$ (Set $x_{ij_i} = 1$);
Let $MC(F_o) = \sum_{j \in F_o} (f_j + Cost\_local\_facility(F_0, V_j, k_j, \infty, 0)) + 2 \sum_{i \in D} (w_C d_{ij_i} + w_L d^L_{j_i} \chi) \lambda_i$;
If $min\_cost - MC(F_o) > 0$
  Let $new\_cost = min\_cost - MC(F_o)$
  For $k_C = \lceil \frac{\lambda_C}{\mu} \rceil$ to $\lfloor \frac{min\_cost - MC(F_o)}{s_C} \rfloor$ do
    For $V_C = \lfloor \frac{min\_cost - MC(F_o) - k_C s_C}{S_C} \rfloor$ downto 0 do
      $new\_cost := \min\{new\_cost, \sum_{j \in F_o} (Cost\_local\_facility(F_o, V_j, k_j, V_C, k_C) + S_C V_C + s_C k_C)\}$
Return $\sum_{j \in F_o} f_j + new\_cost + 2 \sum_{i \in D} (w_C d_{ij_i} + w_L d^L_{j_i} \chi) \lambda_i$

Next we present the optimization procedure of the stock and servers at a local facility location.

$Cost\_local\_facility(F_o, V_j, k_j, V_C, k_C)$
Let $budget = min\_cost - (\sum_{j \in F_o} f_j + S_C V_C + s_C k_C + 2 \sum_{i \in D} (w_C d_{ij_i} + w_L d^L_{j_i} \chi) \lambda_i)$
Let $min\_ser\_cost = \lfloor \frac{budget - \lceil \frac{\sum_{i \in D} \lambda_i x_{ij}}{\mu} \rceil s_j}{S_j} \rfloor S_j + \lfloor \frac{budget}{s_j} \rfloor s_j$
  For $V_j = 0$ to $\lfloor \frac{budget - \lceil \frac{\sum_{i \in D} \lambda_i x_{ij}}{\mu} \rceil s_j}{S_j} \rfloor$ do
    For $k_j = \lceil \frac{\sum_{i \in D} \lambda_i x_{ij}}{\mu} \rceil$ to $\lfloor \frac{budget}{s_j} \rfloor$ do
      If $Fillrate(V_j, k_j, \sum_{i \in D} \lambda_i x_{ij}, V_C, k_C) \geq \alpha$
        $min\_ser\_cost := \min\{min\_ser\_cost, s_j k_j + S_j V_j\}$
Return min_ser_cost

4 Computational study

This section is organized as follows. After presenting an effcient way for calculating the fillrate, we will describe how we constructed the test instances and we will conclude with the results obtained by the algorithm we proposed.

4.1 Some implementation issues

We have seen in Section 2.1 that a straightforward calculation of the fillrate involves a large number of operations, resulting in a very slow algorithm. The running time of the algorithm decreases considerably if one makes use of recursive relations derived from the expression of the fillrate.

The approximation we have made in Section 2.1 implies that for each facility j, the variables N_j, T_j and B_{Cj} are independent. Hence, the fillrate can be calculated by the formula

$$\mathrm{P}(N_j + T_j + B_{Cj} < V_j) = \sum_{k=0}^{k_j} \sum_{i=0}^{V_j - k} \mathrm{P}(N_j = k)\mathrm{P}(B_{Cj} = i)P(T_j < V_i - k) \qquad (6)$$
$$+ \mathrm{P}(N_j > k_j, N_j + T_j + B_{Cj} < V_j)$$

In this formula, several quantities can be calculated recursively as follows. First, based on the known results about a $M/M/k_j$, $P(N_j = k)$, $k = 0, ...$ satisfy:

$$\mathrm{P}(N_j = k+1) = \begin{cases} \mathrm{P}(N_j = k)\frac{\lambda_j}{(k+1)\mu_j} \text{ for } k = 0, \cdots, k_j - 1 \\ \mathrm{P}(N_j = k)\frac{\lambda_j}{k_j \mu_j} \text{ for } k = k_j, k_j + 1, \cdots, \end{cases}$$

where

$$P(N_j = 0) = \left(\sum_{i=0}^{k_j - 1} \frac{(k_j \rho_j)^i}{i!} + \frac{(k_j \rho_j)^k}{k_j!} \frac{1}{1 - \rho_j} \right)^{-1}.$$

Another recursion we use regards $R(V_j, k_j) = \mathrm{P}(N_j > k_j, N_j + T_j + B_{Cj} < V_j)$. These quantities satisfy

$$\begin{aligned} R(V_j + 1, k_j) &= \sum_{k=k_j+1}^{V_j+1} \mathrm{P}(N_j = k)\mathrm{P}(T_j + B_{Cj} < V_j + 1 - k) \\ &= \sum_{k=k_j+1}^{V_j+1} \frac{\lambda_j}{k_j \mu_j} \mathrm{P}(N_j = k-1)\mathrm{P}(T_j + B_{Cj} < V_j - (k-1)) \\ &= \frac{\lambda_j}{k_j \mu_j} \left(R(V_j, k_j) + \mathrm{P}(N_j = k_j)\mathrm{P}(T_j + B_{Cj} < V_j - k_j) \right) \end{aligned}$$

The last recursion we use, concerns the distribution of B_{Cj}.
Let $k_C^+ = \max(1, k_C - V_C)$ and $\rho_{Cj} = p_j \rho_C/(1-)1 - p_j)\rho_C)$.

Lemma 1 *The distribution of B_{Cj} is given for $k = 0, \cdots, k_C^+$ by*

$$\begin{aligned} P(B_{Cj} = i) &= \sum_{n=0}^{k_C^+ - 1} P(B_C = n) \binom{n}{i} p_j^i (1 - p_j)^{n-i} \\ &\quad + P(B_C \geq k_C^+) \sum_{m=0}^{i} \binom{k_C^+}{m} p_j^m (1 - p_j)^{k_C^+ - m} (1 - \rho_{Cj})(\rho_{Cj})^{i-m}, \end{aligned}$$

and for $i = k_C^+, k_C^+ + 1, \cdots$ by

$$P(B_{Cj} = i) = (\rho_{Cj})^{i - k_C^+} P(B_{Cj} = k_C^+)$$

Proof. Before proving the lemma, note that the backlog at the modified central location B_C satisfies

$$\mathrm{P}(B_C = n) = \begin{cases} \sum_{m=0}^{V_C} \mathrm{P}(N_C = m) & k = 0 \\ \mathrm{P}(N_C = n + V_C) & n = 1, \cdots, k_C^+ \\ \mathrm{P}(B_C = k_C^+)\rho_C^{k - k_C^+} & n = k_C^+, k_C^+ + 1. \cdots \end{cases}$$

The last relation follows from $\mathrm{P}(B_C = n) = \mathrm{P}(N_C = n + V_C) = \rho_C^{n+V_C-k_C}\mathrm{P}(N_C = k_C)$ for $k = k_C^+, k_C^+ + 1, \cdots$.

First consider the case $i \le k_C$. Note that, in the modified system, at most k_C^+ servers can be busy at the central facility. The probability that exactly n servers are busy equals $\mathrm{P}(B_C = n)$ for $n = 0, \cdots, k_C^+ - 1$ and equals $\mathrm{P}(B_C \ge k_C^+)$ for $n = k_C^+$. By conditioning on the number of busy servers, we obtain that

$$
\begin{aligned}
Pr(B_{Cj} = i) &= \sum_{n=0}^{k_C^+-1} \mathrm{P}(B_{Cj} = i | B_C = n)\mathrm{P}(B_C = n) + \mathrm{P}(B_{Cj} = i | B_C \ge k_C^+)\mathrm{P}(Bc \ge k_C^+) \\
&= \sum_{n=0}^{k_C^+-1} \binom{n}{i} p_j^i (1-p_j)^{n-i}\mathrm{P}(B_C = n) + \mathrm{P}(B_{Cj} = i | B_C \ge k_C^+)\mathrm{P}(Bc \ge k_C^+), \quad (7)
\end{aligned}
$$

where $p_j = \lambda_j / \lambda_C$.

Denote by S_{Cj} the number of items coming from location j that are being served. Then,

$$\mathrm{P}(B_{Cj} = i | B_C \ge k_C^+) = \sum_{m=0}^{i} \mathrm{P}(B_{Cj} = i | S_{Cj} = m, B_C \ge k_C^+)\mathrm{P}(S_{Cj} = m | B_C \ge k_C^+). \quad (8)$$

Clearly,

$$\mathrm{P}(S_{Cj} = m | B_C \ge k_C^+) = \binom{k_C^+}{m} p_j^m (1-p_j)^{k_C^+-m}. \quad (9)$$

Note that $B_{Cj} - S_{Cj}$ is the number of items coming from location j that are waiting at the central facility. The probability that exactly ℓ items coming from location j are waiting, given that all servers are busy, can be found by conditioning on the total number of waiting items, as follows:

$$
\begin{aligned}
\mathrm{P}(B_{Cj} - S_{Cj} = l | B_C \ge k_C^+) &= \sum_{n=\ell}^{\infty} \mathrm{P}(B_C = k_C^+ + n | B_C \ge k_C^+) \binom{n}{\ell} p_j^\ell (1-p_j)^{n-\ell} \\
&= \sum_{n=\ell}^{\infty} \rho_C^n (1-\rho_C) \binom{n}{\ell} p_j^\ell (1-p_j)^{n-\ell} = \frac{(p_j\rho_C)^\ell (1-\rho_C)}{1-\rho_C(1-p_j)} \\
&= (\rho_{Cj})^\ell (1-\rho_{Cj}). \quad (10)
\end{aligned}
$$

By combining relations (7)-(10) we obtain the first part of the Lemma.

For $i = k_C^+, k_C^+ + 1, \cdots$ we compare the probabilities $\mathrm{P}(B_{Cj} = i)$ with $\mathrm{P}(B_{Cj} = k_C^+)$. First note that when $B_{Cj} \ge k_C^+$, all servers at the central facility are busy. It follows that

$$\mathrm{P}(B_{Cj} = k_C^+) = \mathrm{P}(B_C \ge k_C^+) \sum_{m=0}^{k_C^+} \binom{k_C^+}{m} p_j^m (1-p_j)^{k_C^+-m} (1-\rho_{Cj})(\rho_{Cj})^{k_C^+-m}$$

and

$$\mathrm{P}(B_{Cj} = i) = \mathrm{P}(B_C \ge k_C^+) \sum_{m=0}^{k_C^+} \binom{k_C^+}{m} p_j^m (1-p_j)^{k_C^+-m} (1-\rho_{Cj})(\rho_{Cj})^{i-m}.$$

Now it is readily seen that $\mathrm{P}(B_{Cj} = i) = (\rho_{Cj})^{i-k_C^+}\mathrm{P}(B_{Cj} = k_C^+)$. ■

4.2 Computational results

In all instances, there are 50 clusters of demand points and respectively 10, 14, 20 possible locations for opening facilities. The demand points and the facilities are uniformly distributed in the square $[-1, 1] \times [-1, 1]$. The distances correspond to the usual Euclidian distances in the plane. The costs for opening the facilities are uniformly distributed in [30,90]. The cost of service at each facility j is uniformly distributed in $[10, 100 - f_j]$ and the cost of an item is $0.25 * f_j$. The repair time is uniformly distributed in $[1/10, 2/10]$ and the fill rate is 0.95. The cost of a server is 40, the transportation costs of an item from a demand cluster to a local facility is 10 per unit of distance and from a local facility to the central facility is 40 per unit of distance. In the first type of instances the repair time is 0.9, the cost of an item is 1.5 and the repair probability is 0.95. In the second type of instances, the repair time is 0.1, the cost of an item is 15 and the repair probability is 0.85. In the second case, more items will be sent to the central facility. For all these instances we found that the local search procedure finds the optimal solution.

As a third type, we choose the facility costs uniformly distributed in [30000, 90000]. The other costs are the same as in the first two types. The big difference between the costs for opening facilities and the other costs reduces the problem to a hard facility location problem (see [4]). In this case the local search procedure performs not as well as for the other types, and improved searching techniques have to be used.

5 Concluding remarks

In this paper we have proposed a heuristic for a logistic problem that combines facility location with spare parts management. The computational results obtained show that the heuristic works well in practice. There are many interesting questions raised by the presented problem. One of the assumptions made in this article is that, at an open location, one can install as many servers as needed in order to handle the demand. However, in many practical situations, one can install only a limited number of servers, due to budget constraints. Another assumption is that customers go to the nearest open facility. Can we do better by assigning them to a more distant open faciltiy with some spare capacity. Many variants are possible and can be, stochastically spoken, analyzed by the method described in this paper. Another alternative would be to model the repairshops not as one multiserver queue, but as a jobshop. This is done in Avsar and Zijm; however the recursions given in section 4.1 are more difficult, if not impossible, to find. As a consequence, the computation time will increase considerably.

References

[1] Aardal, K., Reformulation of capacitated facility location problems: how redundant information can help. *Ann. Oper. Res.* 82 (1998), 289–308.

[2] Aardal, K., Capacitated facility location: separation algorithms and computational experience. *Math. Programming* 81 (1998), no. 2, Ser. B, 149–175.

[3] Avsar, Z.M. and Zijm, W.H.M., Capacitated Two-Echelon Inventory Models for Repairable Items Systems, In: S.B. Gershwin et al. (eds.): Analysis and Modeling of Manufacturing Systems, Kluwer Academic Publishers, Boston, ISBN 1-4020-7303-8, 2002.

[4] Barahona F., and Chudak, F., Solving large scale uncapacitated facility location problems. In Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems, P. M. Pardalos, Editor, Kluwer Academic Publishers, June, 1999.

[5] Batta, R., Single server queuing-location models with rejection, *Transportation Science* 22 (1988) 209-216.

[6] Batta, R., A queuing location model with expected service time dependent queuing disciplines, *European Journal of Operational Research* 39 (1989) 192-205.

[7] Beasley, J.E., Lagrangean heuristics for location problems. *European Journal of Operational Research* 65, (1993) pages 383-399.

[8] Berman, O., Larson, R. and Chiu, S.,Optimal server location on a network operating as a M/G/1 queue, *Operations Research* 12 (1985) 746-771.

[9] Berman, O., Larson, R. and Parkan C., The stochastic queue p-median location problem, *Transportation Science* 21 (1987) 207-216.

[10] Berman, O. and Mandowsky, R., Location-allocation on congested networks, *European Journal of Operational Research* 26, (1986) 238-250.

[11] Christofides, N. and Beasley, J.E., Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem, *European Journal of Operational Research* 12, (1983), pages 19-28.

[12] Cornuejols, G., Sridharan, R. and Thizy, J., A comparison and relaxations for the capacitated plant location problem, *European Journal of Operational Research* 50 (1991) 280-197.

[13] Davis, P. and Ray, T., A branch and bound algorithm for the capacitated facility location problem, *Naval Research Logistic Quarterly* 16 (1969) pages 331-334.

[14] Drezner, Z., editor. Facility Location: A Survey of Applications and Methods,Springer-Verlag New York,1995.

[15] Hesse Owen, S. and Daskin, M.S., Strategic facility location: a review. *European Journal of Operational Research* 111 (1998) pages 423-447.

[16] Kuehn, A. A. and Hamburger, M.J. A heuristic program for locating warehouses. *Management Science*, 9 (1963) 643-666.

[17] Larson, R.C., A hypercube queueing model for facility location and redestricting in urban emergency services, *Computers and Operations Research* 1 (1974) 67-95.

[18] Marianov, V. and Serra, D., Probabilistic, maximal covering location-allocation models for congested systems, *Journal of Regional Science* 38 (1998) 401-424.

[19] Marianov, V. and Serra, D., Location-allocation of Multiple-server service centers with constrained queues or waiting tomes, *Annals of Operations Research* 111, 35-50, 2002

[20] Mirchandani, P. and Francis, R. editors. Discrete Location Theory. John Wiley and Sons, Inc, New York, 1990.

[21] Muckstadt, J.A., A Model for a Multi-Item Multi-Echelon Multi-Indenture Inventory System. *Management science*, vol 20, pp. 472-481, 1973.

[22] Teitz, M.B. and Bart, P., Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research* , 16 (1968) 955-961.

[23] ReVelle, C., Facility sitting and integer-friendly programming, *European Journal of Operational Research* 65, (1993) 147-158.

[24] Sherbrooke, C.C., METRIC: A Multi-Echelon Technique for Recoverable Item control. *Operations Research*, vol 16, pp.122-141, 1968.

[25] Sleptchenko, A., Integral Inventory Control in Spare Parts Networks with Capacity Restrictions, PhD thesis, University of Twente, 2002.

[26] Wang, Q., Batta, R. and Rump, C., Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of Operations Research* 111, 17-34, 2002