Locating repair shops in a stochastic environment

J.C.W. van Ommeren*, A.F. Bumb
Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands,

A.V. Sleptchenko
EURANDOM and Department of Technology and Management,
Eindhoven University of Technology, The Netherlands

December 15, 2004

Abstract

In this paper we consider a repair shop location problem with uncertainties in demand. New 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, an 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 we focus on, is finding locations for the local repair shops, deciding their capacity, i.e., number of servers and base stock levels, such that the total expected cost is minimized and the fraction of customers that can leave the local shops without waiting is above some specified level. We assume that the central repair shop is already opened, but that the repair capacity still has to be set. The costs we consider are the costs for keeping the repair shops operational, for the transport of items and for the inventory. For this problem, a local search heuristic is proposed and experimental results are presented.

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

*Corresponding author, e-mail address: j.c.w.vanommeren@utwente.nl

1 Introduction

In this paper we present a model for the following problem. Consider a transport company, which uses buses/trucks for transportation of passengers/goods. The company decides to build small repair shops for the maintenance of the vehicles. When a vehicle is defective, its driver brings it to the repair shop, where the broken part is replaced in a neglectable time. The broken part is repaired, in principal, at the local repair shop. If the broken part, for some reason, can not be repaired at the local repair shop, it is sent to the central repair shop. There are several locations where the local repair shops can be placed. At a repair shop several repair facilities can be installed and spare parts can be kept in inventory in order to insure a high service level. The position of the central repair shop is known in advance, but the number of servers that will be installed there is not. When deciding if at a certain location a repair shop 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 customers (each customer is 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 addressing 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 [22] and the review done by Hesse, Owen and Daskin [15].

Recently, the issue of uncertainty of demand and transportation has been 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 (Batta [5, 6], Berman, Larson and Chiu [8], Larson [18]). In the case of more mobile servers, the algorithms developed generally use as a sub-algorithm the single-server model (Berman, Larson and Parkan [9], Berman and Mandowsky[10]). Marianov and Serra [19] analyse the issue of locating servers at fixed locations when the number of requests for service follow some probabilistic distribution. Their goal is 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 Marianov and Serra [21] they extend their analysis to the situation in which the number of facilities and servers needed to cover all the population is minimized. In [28], 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 cost. In their model there is an upper bound on the number of open facilities and on the expected waiting time at a facility.

For literature on spare part management, we refer to Sherbrooke [26], Muckstadt [23], Avsar and Zijm [4] and Sleptchenko [27]. 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. Avsar and Zijm model the system as a product form network, by assuming that the repair times are exponentially distributed while Sleptchenko gives an approximation based on the first two moments of the repair times. They use the analytic results to find optimal inventory levels.

This 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 repair shops as M/M/k queues and consider deterministic transportation times. The quality of service is measured by the 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 Avsar and Zijm in [4]. The approximation of the fill rate and the calculation of the expected inventory levels at facilities are presented in Section 3. In Section 4 we propose a local search heuristic for finding a solution of the problem. In Section 5 we present computational results illustrating the behavior of the proposed procedure. The numerical results obtained by the local search procedure are compared with a brute force procedure and a procedure based on sequential optimization. We present our conclusions in Section 6.

2 The model

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

At each local repair shop a stock of items is kept in order to replace the broken items brought by customers. The broken items that have been repaired locally, are put in stock and are ready to use. By some external cause, broken items sometimes can not be repaired locally. These items are sent for repair 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 repair shop, 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 for every item transportation to the central repair shop is available when is needed. At all repair shops, both local and central, several servers can be installed. Arriving requests which cannot be served immediately, are put in a queue (backordered). At the local repair shops, these so called backorders, are of course undesirable. The probability that a customer does not have to wait for service is called the *fill rate*.

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

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

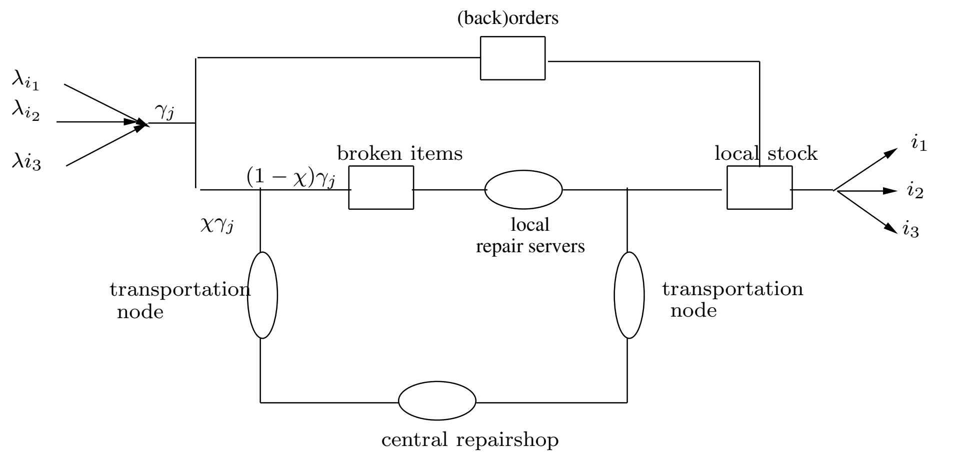

Figure 1: Local repair shop j

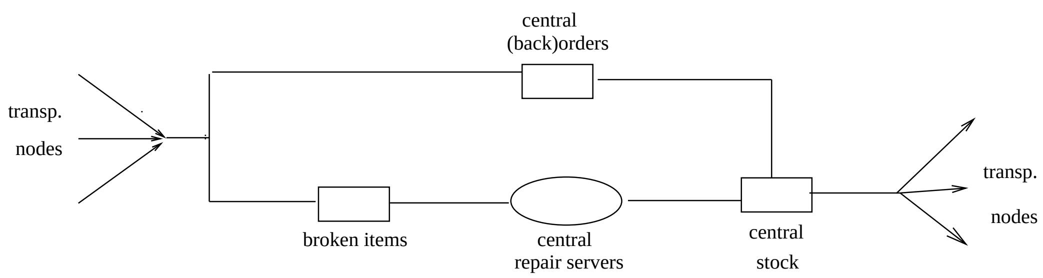

Figure 2: Central repair shop

One has to decide where to open repair shops (facilities), how many repair servers to install and what the base stock levels should be, in order to insure the specified fill rate at the lowest expected cost. The costs we consider are related to the stock levels, the opening of local repair shops, the installation of repair servers, transportation from customers to local repair shops, transportation from the local repair shops to the central facility and vice versa. All the transportation costs are considered proportional to the distances.

To model this situation, we introduce the following notations:

Inputs and parameters

- D: set of N clusters of customers (demand points);
- F: set of M locations where local repair shops can be opened;
- d_{ij}: time units needed for customers in cluster i to reach location j;
- d_j^L: time needed to reach the central repair shop from a local repair shop j;
- f_j: amortization (over time) of the cost of opening a repair shop at location j;
- h_j: unit cost (per unit time) of holding stock at location j;
- h_C: unit cost (per unit time) of holding stock at the central location;
- s_C: server cost per unit time at the central facility;

s_j: server cost per unit time at location j;

w_L: the unit cost for the internal transportation from the local repair shops to the central repair shop;

w_D: the external transportation cost of customers;

χ: the probability that a broken item cannot be repaired at a local repair shop and is sent for repair to the central one;

λ_i: the rate at which requests for repair are generated at cluster i;

γ_C: the arrival rate at the central repair shop, i.e., $\gamma_C = \chi \sum\limits_{i \in D} \lambda_i$;

μ_j: service rate of a single server at local repair shop j;

μ_C: service rate of a single server at the central repair shop;

α: the prescribed minimal value of the fill rate.

Decision variables

y_j: variable indicating whether a repair shop at location $j \in F$ is open

x_{ij}: variable indicating if cluster i of demand points goes to repair shop $j \in F$.

V_j: base stock level at local repair shop $j \in F$;

V_C: base stock level at the central repair shop;

k_j: number of servers at local repair shop $j \in F$;

k_C: number of servers at the central repair shop;

γ_j: the rate at which requests for repair arrive at a local repair shop j;

For simplicity of the presentation, assume that $d_{ij} \neq d_{ik}$ for $j \neq k, j, k \in F$ and $i \in D$. Since each cluster of clients is assigned to the closest open repair shop, 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

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

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

Remark 1 *We assume that an item can not be repaired at a local repair shop by some external cause, for instance, if the item is severely damaged. This assumption implies that χ, the probability that an item can not be repaired locally, is an input parameter and has to be specified. Another model can be developed if the value of χ depends on the available equipment (lower χ corresponding to higher server costs). In this case, χ would be a decision variable.*

We are interested in the inventory holding costs and the fill rates as functions of the decision variables. Denote the actual local and central inventory levels by I_j and I_C. To analyse these inventory levels and fill rates, we introduce the following stochastic model.

A repair shop at location j can be opened by installing $k_j > 0$ servers. We assume that these installed servers have exponential distributed service times with expectation $1/\mu_j$. The requests originating from demand point i with $i \in D$ form independent Poisson processes with rate λ_i, respectively. The arriving requests at an open repair shop at location j are a combination of a number of independent Poisson processes and form, therefore, a Poisson process with arrival rate $\gamma_j = \sum_{i \in D} \lambda_i x_{ij}$. Arriving items are, independently, locally repairable with probability χ. Thus, both the input to the servers and the items sent from location j to the central location are filtered Poisson processes with rate $(1-\chi)\gamma_j$ and $\chi\gamma_j$, respectively. In order to satisfy the stability requirements, we impose, at every open repair shop, an arrival rate smaller than the service rate. In other words, for each $j \in F$,

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

Similarly, the total arrival rate at the central repair shop is $\chi \sum_{i \in D} \lambda_i = \gamma_C$. In order to have stability at the central repair shop, the following relation should hold

$$\gamma_C < k_C \mu_C.$$

In the next section, we use this stochastic model to analyse the behavior of the inventory levels and fill rates.

Remark 2 *For the stochastic modeling of the repair shops we make two major assumptions. The assumption that the arrival of requests form a Poisson process, is often justified by the data. In many practical situations, where the first moment of the service times is already hard to come by, the choice of exponentially distributed service times is very common. Although the exponential distribution may not give the most accurate description, it enables the analysis of the model.*

The total expected cost (the cost of opening facilities, the expected inventory costs and the expected transportation costs) per unit time can be written as

$$\sum_{j \in F}(f_j + k_j s_j)y_j + h_C E(I_C) + s_C k_C + \sum_{j \in F} h_j E(I_j) y_j + 2\sum_{j \in F}\sum_{i \in D}(w_D d_{ij} + w_L d_j^L \chi)\lambda_i x_{ij},$$

where $E(I_C)$ and $E(I_j)$ are the expected inventory levels at the central repair shop, respectively local repair shop j. These expectations are functions of the number of servers and the base stock levels. The expected transportation costs are found by applying Little's Law (cf. Ross [25]). A detailed description of how the expected values appearing in the objective function can be calculated will be given in next section.

For covering the situation where the "customers" are a part of the company (see the example in the introduction), we include the transportation costs for the customers in the objective function. If the amounts that customers have to pay for transportation, do not play a role in the decision process, they may be neglected. The analysis of the model and the proposed heuristic will remain unchanged.

We arrive at the following mathematical programming formulation:

$$\min \sum_{j\in F}(f_j + s_jk_j)y_j + h_C E(I_C) + s_C k_C + \sum_{j\in F} h_j E(I_j) y_j$$
$$+ 2\sum_{j\in F}\sum_{i\in D}(w_D d_{ij} + w_L d_j^L \chi)\lambda_i x_{ij}$$

$$s.t. \sum_{j\in F} x_{ij} = 1, \text{ for each } i \in D \tag{2}$$

$$x_{ij} \le y_j, \text{ for each } i \in D \text{ and } j \in D \tag{3}$$

$$\sum_{k\in F} d_{ik}x_{ik} \le (d_{ij} - \Delta)y_j + \Delta, \text{ for each } i \in D \text{ and } j \in F \tag{4}$$

$$(1-\chi)\sum_{i\in D}\lambda_i x_{ij} < \mu_j k_j y_j, \text{ for each } j \in F \tag{5}$$

$$\gamma_C < k_C \mu_C \tag{6}$$

$$Fillrate(V_j, k_j, \sum_{i\in D}\lambda_i x_{ij}, V_C, k_C) \ge \alpha y_j, \text{ for each } j \in F \tag{7}$$

$$y_j \in \{0,1\}, \text{ for each } j \in F$$

$$k_j, k_C, V_j, V_C \in Z_+, \text{ for each } j \in F$$

Constraints (2) and (3) insure that each cluster of customers is assigned to an open local repair shop, (4) insures that each cluster of customers is assigned to the closest open repair shop, (5), respectively (6) insure stability at each local, respectively central repair shop and (7) insures the required quality of service at an open repair shop. Note that without constraint (7), the problem reduces to a variant of the capacitated facility location problem, in which inventory is also taken into account. However, the intractability of constraint (7) increases considerably the degree of difficulty of the described problem. In the next section we will derive an approximation of the fill rate, that can be easily described analytically.

3 The fill rate and the expected inventory level

In our model, we are interested in the steady state behavior of the system, therefore we will omit the time dependence of all the stochastic variables. Throughout the remainder of the paper, we will use 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 repair shop $j \in F$ to the central repair shop and vice versa;

B_C : number of backorders at the central repair shop, i.e., $B_C = max\{N_C - V_C, 0\}$.

B_{Cj} : number of backorders at the central repair shop originating at location $j \in F$. Note that $B_C = \sum_{j\in F} B_{Cj}$.

The fill rate at location j, i.e., the probability that a customer does not have to wait for service at a location j, follows from the PASTA property (Wolff [29]) and is equal to

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

If we knew the distribution of $((N_j, T_j, B_{Cj}), j = 1, ..., N)$, we could easily calculate the fill rate. However, finding this distribution turns out to be a difficult task, since, on the one hand, due to the deterministic transportation times, the vector process is not Markovian, and on the other hand, due to the inventory at the central repair shop, the network has no product form. If the transportation times were exponentially distributed, the process $((N_j, T_j, N_C), j = 1, ..., N)$ viewed over time, would indeed be a Markov process, and the fill rate could, in principal, be computed. However, it would still be intractable due to the large state space.

As an alternative, we use the product form approximation described in Avsar and Zijm [4] for the distribution of $((N_j, T_j, B_{Cj}), j = 1, ..., N)$. They show that the network has a product form if one replaces the central facility, with its base stock policy, by a special state dependent server (see Figure 3).

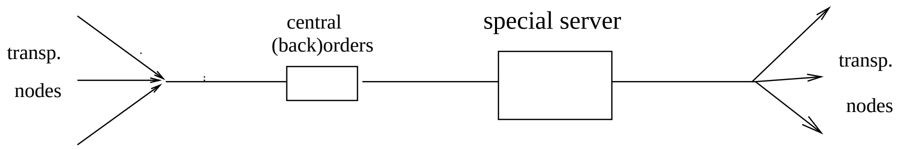

Figure 3: Central facility with a special server

The server works at speed $min\{V_C + k, k_C\}\mu_C$, whenever there are $k > 0$ backordered 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 central facility immediately, otherwise it is served with finite speed. The probability that an item is served at infinite speed can be interpreted, in the original model, as the probability that a replacement is available when an item arrives to find no backorders at the central facility. By choosing q in this way, the number of backlogged items in the original system and the modified system, have the same steady state distribution as it is shown in [4]. From now on, we do not distinguish the number of backlogged items in the original system (B_C) and in the modified system ($\overline{B_C}$). The steady state distribution of the number of backordered items at the central repair shop is given by

$$\mathrm{P}(\overline{B_C} = k) = \mathrm{P}(B_C = k) = \begin{cases} \mathrm{P} N_C \leq V_C & k = 0 \\ \mathrm{P}(N_C = V_C + k) & k = 1, 2, \cdots \end{cases} \quad (8)$$

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

In the following, suppose that the central facility is replaced by the special server as proposed in Avsar and Zijm [4]. In this modified model, the steady state distribution has a product form, hence the stochastic variables N_j, T_j and B_{Cj}, $j = 1, ..., N$ can be treated separately. Next we will discuss the distributions of each of these variables, and then we will present two ways of calculating the fill rate.

For each local repair shop j, the random variable N_j is just the number of customers at an $M/M/k_j$ queue with arrival rate γ_j and mean service time $1/\mu_j$.

By definition, T_j is composed of two streams of items: items being transported from repair shop j to the central facility and items being transported from the central facility to repair shop j. From repair shop j, items depart towards central facility according to a Poisson process with rate $\chi\gamma_j$. Hence, the number of items in transportation is Poisson distributed with expectation $\chi\gamma_j d_j^L$. From the product form of the network follows that the number of items in transportation in the two streams are independent, and therefore the number of traveling items is Poisson distributed with expectation $\tau_j = 2\chi\gamma_j d_j^L$, i.e.,

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

Finally, the steady state distribution of B_{Cj}, the number of backorders at the central repair shop originating at location j, can be calculated as follows:

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

From the viewpoint of j, two types of items arrive at the central facility, one originating at the local repair shop j and the other originating at the rest of the local repair shops. Denote by p_j the probability that an arrival at the special server is an item from j. Clearly,

$$p_j = \chi\lambda_j / \chi\gamma_C = \gamma_j / \gamma_C.$$

Moreover, due to the independence of the Poisson arrivals from different local repair shops, the probability that an item backordered at the central repair shop came from repair shop j, is p_j. Hence,

$$\mathrm{P}(B_{Cj} = i | B_C = n) = \binom{n}{i} p_j^i (1 - p_j)^{n-i}.$$

Based on the distributions of N_j, T_j and B_{Cj}, the fill rate at location j can be calculated as follows:

$$\mathrm{P}(N_j + T_j + B_{Cj} < V_j) = \sum_{r=0}^{V_j - 1} \sum_{s=0}^{r} \sum_{q=0}^{r-s} \mathrm{P}(N_j = s)\mathrm{P}(T_j = q)\mathrm{P}(B_{Cj} = r - s - q). \qquad (9)$$

Note that this direct calculation of the fill rate is not very efficient for implementations, due to the large number of operations involved. Next we will present a set of recursions that considerably improve the time needed to calculate the fill rate.

3.1 A faster method of calculating the fill rate

Due to the independence of the variables N_j, T_j and B_{Cj}, the fill rate at facility j can be calculated by the formula

$$\begin{aligned} \mathrm{P}(N_j + T_j + B_{Cj} < V_j) = &\sum_{k=0}^{k_j} \mathrm{P}(N_j = k)\mathrm{P}(B_{Cj} + T_j < V_j - k) \\ &+ \mathrm{P}(N_j > k_j, N_j + T_j + B_{Cj} < V_j). \end{aligned} \qquad (10)$$

In this formula, several quantities can be calculated recursively, as will be described below.

- *Recursion formula for* $P(N_j = k)$

Based on the known results about an $M/M/k_j$ queue, $P(N_j = k)$, $k = 0, ...,$ satisfy (see Ross [25]):

$$\mathrm{P}(N_j = k+1) = \begin{cases} \mathrm{P}(N_j = k)\frac{k_j\rho_j}{k+1} \text{ for } k = 0, \cdots, k_j - 1 \\ \mathrm{P}(N_j = k)\rho_j \text{ for } k = k_j, k_j+1, \cdots, \end{cases} \tag{11}$$

where

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

and

$$\rho_j = \frac{(1-\chi)\gamma_j}{k_j\mu_j}.$$

Note that a similar recursion formula holds for $P(B_C = k)$ (see (8)).

- *Recursion formula for* $\mathrm{P}(N_j > k_j, N_j + T_j + B_{Cj} < V_j)$

For calculating recursively $R(V_j, k_j) = \mathrm{P}(N_j > k_j, N_j + T_j + B_{Cj} < V_j)$, we proceed as follows:

$$\begin{aligned} R(V_j+1, k_j) &= \sum_{k=k_j+1}^{V_j+1} \mathrm{P}(N_j = k)\mathrm{P}(T_j + B_{Cj} < V_j + 1 - k) \qquad (12) \\ &= \sum_{k=k_j+1}^{V_j+1} \rho_j \mathrm{P}(N_j = k-1)\mathrm{P}(T_j + B_{Cj} < V_j - (k-1)) \\ &= \rho_j \left(R(V_j, k_j) + \mathrm{P}(N_j = k_j)\mathrm{P}(T_j + B_{Cj} < V_j - k_j) \right), \end{aligned}$$

where the second equality follows from (11). Note that both relations (10) and (12) contain $\mathrm{P}(B_{Cj} + T_j < m)$ for $m = 0, ..., V_j$. We will concentrate on the distribution of B_{Cj} in Lemma 3. Subsequently, we describe a recursion for the computation of $\mathrm{P}(B_{Cj} + T_j < m)$.

- *Recursion formula for* $\mathrm{P}(B_{Cj} = k)$

Let

$$k_C^+ = min\{1, k_C - V_C\}$$

and

$$\rho_{Cj} = \frac{p_j\rho_C}{1-(1-p_j)\rho_C}, \text{ where } \rho_C = \frac{\gamma_C}{k_C\mu_C}.$$

Lemma 3 *The distribution of* B_{Cj} *is given for* $i = 0, \cdots, k_C^+$ *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. We start with the case where $i = k_C^+, k_C^+ + 1, \cdots$. Note that when $B_{Cj} \geq k_C^+$, the server at the central facility works at full speed $k_C \mu_C$. It follows that

$$\begin{aligned}
\mathrm{P}(B_{Cj} = k) &= \sum_{m=k}^{\infty} \mathrm{P}(B_{Cj} = k | B_C = m)\mathrm{P}(B_C = m) = \sum_{m=k}^{\infty} \binom{m}{k} p_j^k (1-p_j)^{m-k} \mathrm{P}(B_C = m) \\
&= \sum_{m=k}^{\infty} (\binom{m-1}{k-1} + \binom{m-1}{k}) p_j^k (1-p_j)^{m-k} \mathrm{P}(B_C = m) \\
&= p_j \sum_{m=k}^{\infty} \binom{m-1}{k-1} p_j^{k-1} (1-p_j)^{m-1-(k-1)} \mathrm{P}(B_C = m) \\
&\quad + (1-p_j) \sum_{m=k}^{\infty} \binom{m-1}{k} p_j^k (1-p_j)^{m-1-k} \mathrm{P}(B_C = m) \\
&= p_j \rho_C \mathrm{P}(B_{Cj} = k-1) + (1-p_j)\rho_C \mathrm{P}(B_{Cj} = k),
\end{aligned}$$

where we used that $\binom{m}{n} = 0$ for $m < n$ and that $\mathrm{P}(B_C = k+1) = \rho_c \mathrm{P}(B_C = k)$ for $k = k_C^+, k_C^+ + 1, \cdots$. Now it is readily seen that

$$\mathrm{P}(B_{Cj} = k) = \frac{p_j \rho_C}{1 - (1-p_j)\rho_C} \mathrm{P}(B_{Cj} = k-1) = (\rho_{Cj})^{k-k_C^+} \mathrm{P}(B_{Cj} = k_C^+).$$

Consider now the case $i \leq k_C^+$. In the modified system, the server works at different speeds, depending on the number of backlogged items. The probability that the server works at speed $(k_C + n - k_C^+)\mu_C$ equals $\mathrm{P}(B_C = n)$ for $n = 0, \cdots, k_C^+ - 1$ and equals $\mathrm{P}(B_C \geq k_C^+)$ for $n = k_C^+$. By conditioning on the number of backlogged items, we obtain that

$$\begin{aligned}
\mathrm{P}(B_{Cj} = i) &= \sum_{n=0}^{k_C^+-1} \mathrm{P}(B_{Cj} = i | B_C = n)\mathrm{P}(B_C = n) + \mathrm{P}(B_{Cj} = i | B_C \geq k_C^+)\mathrm{P}(B_C \geq k_C^+) \\
&= \sum_{n=0}^{k_C^+-1} \binom{n}{i} p_j^i (1-p_j)^{n-i} \mathrm{P}(B_C = n) + \mathrm{P}(B_{Cj} = i | B_C \geq k_C^+)\mathrm{P}(B_C \geq k_C^+). \quad (13)
\end{aligned}$$

Denote by S_{Cj} the number of items coming from location j that are in the first k_C^+ places of the queue at the central facility. Then

$$\mathrm{P}(B_{Cj} = i | B_C \geq k_C^+) = \sum_{m=0}^{i} \mathrm{P}(B_{Cj} = i | S_{Cj} = m, B_C \geq k_C^+)\mathrm{P}(S_{Cj} = m | B_C \geq k_C^+). \quad (14)$$

Clearly,

$$\mathrm{P}(S_{Cj} = m | B_C \geq k_C^+) = \binom{k_C^+}{m} p_j^m (1-p_j)^{k_C^+ - m}. \quad (15)$$

Note that $B_{Cj} - S_{Cj}$ is the number of items coming from location j, 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:

$$\begin{aligned}
\mathrm{P}(B_{Cj} - S_{Cj} = \ell | B_C \geq k_C^+) &= \sum_{n=\ell}^{\infty} \mathrm{P}(B_C = k_C^+ + n | B_C \geq k_C^+) \binom{n}{\ell} p_j^{\ell} (1-p_j)^{n-\ell} \\
&= \sum_{n=\ell}^{\infty} \rho_C^n (1-\rho_C) \binom{n}{\ell} p_j^{\ell} (1-p_j)^{n-\ell} = \frac{(p_j \rho_C)^{\ell} (1-\rho_C)}{[1-\rho_C(1-p_j)]^{l+1}} \\
&= (\rho_{Cj})^{\ell} (1-\rho_{Cj}). \qquad (16)
\end{aligned}$$

By combining relations (13)-(16) we obtain the first part of the lemma. ■

In the next paragraph we present a recursion scheme for the computation of $\mathrm{P}(B_{Cj}+T_j < m)$.

• *Recursion formula for* $\mathrm{P}(B_{Cj} + T_j < m)$

By splitting up the event $B_{Cj} + T_j < m + 1$ in disjoint subsets according to the value of B_{Cj}, we get:

$$\begin{aligned}
\mathrm{P}(B_{Cj} + T_j < m + 1) = &\sum_{i=0}^{k_C^+} \mathrm{P}(B_{Cj} = i) \mathrm{P}(T_j < m + 1 - i) \\
&+ \mathrm{P}(B_{Cj} > k_C^+, B_{Cj} + T_j < m + 1).
\end{aligned}$$

The previous lemma now gives (cf. formula (12)) that

$$\begin{aligned}
\mathrm{P}(B_{Cj} > k_C^+, B_{Cj} + T_j < m + 1) = &\rho_{Cj} \mathrm{P}(B_{Cj} > k_C^+, B_{Cj} + T_j < m) \qquad (17) \\
&+ \rho_{Cj} \mathrm{P}(B_{Cj} = k_C^+) \mathrm{P}(T_j < m - k_C^+). \qquad (18)
\end{aligned}$$

The last recursion scheme we give regards $\mathrm{P}(T_j < k + 1)$.

• *Recursion formula for* $\mathrm{P}(T_j < k + 1)$

Note that $\mathrm{P}(T_j < k + 1)$ can be also calculated recursively as $\mathrm{P}(T_j < k + 1) = \mathrm{P}(T_j < k) + \mathrm{P}(T_j = k)$. Since T_j has a Poisson distribution with parameter $\tau_j = 2\chi\gamma_j d_j^L$,

$$\mathrm{P}(T_j = k) = \frac{\tau_j}{k} \mathrm{P}(T_j = k - 1) \text{ for } k = 1, 2, \ldots$$

3.2 Expected inventory levels

In this subsection we concentrate on the expected inventory levels at the facilities. These quantities are needed to calculate the total expected cost of a solution with given base stock levels and numbers of servers.

For the expected inventory level at the central facility, we have to return to the original model, since, in the modified system, we only focus on the number of backlogged items, which is positive only when the stock is empty. For the expected inventory level at the local facilities, we use the fill rates that were computed in the previous subsection. In both cases, we use that the expectation of a nonnegative discrete random variable L is given by $E(L) = \sum_{k=1}^{\infty} \mathrm{P}(L \geq k)$.

For the expected inventory level at the central facility, remember that $I_C = \max\{0, V_C - N_C\}$. If $V_C \leq k_C$, then

$$E(I_C) = \sum_{k=1}^{V_C} \mathrm{P}(I_C \geq k) = \sum_{k=0}^{V_C-1} \mathrm{P}(N_C < k).$$

Since N_C is the number of items in an $M/M/k_C$ queue, $E(I_C)$ can now easily be computed.

If $V_C \geq k_C$, remark that $I_C = V_C - N_C + \max\{0, N_C - V_C\}$ so

$$E(I_C) = V_C - E(N_C) + E(\max\{0, N_C - V_C\}).$$

To find the expectations in the right hand side, we proceed as follows:

$$\begin{aligned} E(\max\{0, N_C - V_C\}) &= \sum_{k=1}^{\infty} \mathrm{P}(N_C \geq V_C + k) \\ &= \sum_{k=1}^{\infty} \mathrm{P}(N_C = k_c))\rho_C^{V_c - k_C} \frac{\rho_C^k}{1 - \rho_C} \\ &= \mathrm{P}(N_C = k_c)\rho_C^{V_c - k_C} \frac{\rho_C}{(1 - \rho_C)^2}. \end{aligned}$$

The value of $E(N_C)$ can either be found as the expected number of customers in an $M/M/k_C$ queue or by combining the formulas for the two cases $V_C \leq k_C$ and $V_C \geq k_C$.

The expected inventory level at the local facilities are easy to compute once we know the fill rates at these facilities for all base stock levels $k = 1, \ldots, V_j$, namely

$$E(V_j) = \sum_{k=1}^{V_j} \mathrm{P}(I_j \geq k) = \sum_{k=1}^{V_j} \mathrm{P}(N_j + T_j + B_{Cj} < k).$$

4 A local search heuristic

The model presented in the Section 2 is a variant of the capacitated facility location problem with additional nonlinear constraints regarding the fill rates. Even the linear variant of the capacitated facility location problem is considered very difficult (see ReVelle [24] for a discussion on the implication of the capacity constraints). Many methods have been proposed for tackling the linear capacitated facility location problem, such as Lagrangian relaxation (*e.g.* Beasley [7], Christofides, Beasley[11], Cornuejols, Sridharan, Thizy [12]), polyhedral approach Aardal[1], branch and bound (Aardal[2], Davis, Ray [13]) and local search Kuehn, Hamburger [17]. In this paper we opt for a local search approach, based on the procedure developed by Kuehn and Hamburger in [17]. Local search heuristics have proved to give good results for facility location problems, both experimentally and from worst case point of view (Arya et al. [3], Korupolu, Plaxton, Rajaraman [16]).

Suppose for the moment that we have developed a procedure called $Cost\_set\_facilities(S)$ which returns the costs involved when one knows that the facilities in S are open. The local search heuristic then proceeds as follows. Start with a set F_o of open facilities and in each step execute the operation with the largest cost improvement among the following three: open a new facility $i \in F \setminus F_o$, i.e., $F_o := F_o \cup \{i\}$, close an already opened facility $j \in F_o$, i.e., $F_o := F_o \setminus \{j\}$ or swap facilities (open a closed facility $i \in F \setminus F_o$ and close an open facility $j \in F_o$), i.e., $F_o := F_o \cup \{j\} \setminus \{i\}$. The procedure stops when no cost improvement is possible.

The procedure is summarized below.

Local Search Procedure

Start with a set of open facilities F_o
Find the set $\tilde{F_o}$ for which $\underset{\tilde{F_o}:|\tilde{F_o}\setminus F_o|\leq 1}{\text{minimum}}\, Cost\_set\_facilities(\tilde{F_o})$ is attained.
While $Cost\_set\_facilities(\tilde{F_o}) \leq Cost\_set\_facilities(F_o)$
  Replace F_o with $\tilde{F_o}$ ($F_o := \tilde{F_o}$).
Return the set F_o.

Remark 4 *The procedure presented above, is the simplest form of local search. A combination with other methods, e.g. Tabu Search (see Marianov, Serra [20]) or the Heuristic Concentration method (see Marianov, Serra [21]), may improve the results of the algorithm. However, we have chosen for the simple procedure above, since our goal is to show how the analysis of the model can be combined with a procedure for facility location problems which is based on the comparison of the costs of several combinations of open facilities.*

Next we describe the procedure $Cost\_set\_facilities(\tilde{F_o})$, for calculating the cost associated with a set of open facilities $\tilde{F_o}$. Let facility j_i be the closest facility in $\tilde{F_o}$ to client $i \in D$. Assume that we have at disposal a procedure, called $Servers\_Inventory(\tilde{F_o})$, for calculating the inventory and the number of servers with minimal expected costs at the facilities in $\tilde{F_o}$ and at the central facility, which insure the required fill rate. Then $Cost\_set\_facilities(\tilde{F_o})$ consists of the following quantities: the costs of opening the facilities in $\tilde{F_o}$, i.e., $\sum_{j\in\tilde{F_o}} f_j$, the expected transportation costs, i.e., $2\sum_{i\in D}(w_D d_{ij_i} + w_L d^L_{j_i}\chi)\lambda_i$ and the inventory and server costs at the facilities in $\tilde{F_o}$ and at the central facility, i.e., $Servers\_Inventory(\tilde{F_o})$. Denote by $OT(\tilde{F_o})$ the total expected cost associated with $\tilde{F_o}$ for opening the facilities and the transportation, that is

$$OT(\tilde{F_o}) = \sum_{j\in\tilde{F_o}} f_j + 2\sum_{i\in D}(w_D d_{ij_i} + w_L d^L_{j_i}\chi)\lambda_i.$$

Then the total cost associated with $\tilde{F_o}$ is given by

$$Cost\_set\_facilities(\tilde{F_o}) = OT(\tilde{F_o}) + Servers\_Inventory(\tilde{F_o}).$$

The necessary number of servers and the inventory (stock) at facilities in $\tilde{F_o}$ and at the central facility, given by $Servers\_Inventory(\tilde{F_o})$, is decided as follows. For each feasible combination of servers/inventory at the central facility, (k_C, V_C), calculate the combinations $\{(k_j, V_j), j \in \tilde{F_o}\}$, that ensure the required fill rate and have minimal expected costs. For calculating the feasible combinations (k_C, V_C) we proceed as follows.

Let min_cost be the minimum total expected cost associated with a set of open facilities that was analysed so far. Denote by$MC(\tilde{F_o})$ the total expected cost associated with $\tilde{F_o}$ when there are always items in the central stock at zero cost (inventory level $V_C = \infty$ with expected inventory costs 0 and the number of central servers $k_C = 0$ since the inventory is always full). Clearly, $MC(\tilde{F_o})$ is the expected cost for the local servers and inventories and the transportation costs for the case where there are never backlogged items at the central facility. Hence, $MC(\tilde{F_o})$ is a lower bound of the real cost associated with $\tilde{F_o}$. If $min\_cost < MC(\tilde{F_o})$, we stop analysing $\tilde{F_o}$, since $\tilde{F_o}$ will not give us a solution of lower costs than the one of cost min_cost. If $min\_cost > MC(\tilde{F_o})$, we can interpret the quantity $min\_cost - MC(\tilde{F_o})$ as being the available budget for acquiring servers and inventory at the central facility when the facilities in $\tilde{F_o}$ are open. Based on this available budget, we can calculate the maximum number of servers that can be acquired at the central facility, i.e., $\lfloor \frac{min\_cost - MC(\tilde{F_o})}{s_C} \rfloor$ and the maximum stock for a fixed number of servers k_C, i.e., $E(I_C)h_C \leq min\_cost - MC(\tilde{F_o}) - k_C s_C$ (remark that $E(I_C)$ is a function of V_C and k_C). For a given k_C, define the maximal inventory level that keeps the expected cost for the actual inventory below $t \geq 0$ by

$$V_C^{\max}(t) = \max\{V_C | E(I_C) \leq t/h_C\}.$$

The minimal number of servers that should be installed at the central facility is given by $k_C = \lfloor \frac{\gamma_C}{\mu_C} \rfloor + 1$ and the minimal stock level is 0.

Let $Local\_SI(k_C, V_C, Cost\_so\_far, V_j, k_j)$ be the procedure that, for given (k_C, V_C) calculates the cheapest combination of stock and servers (k_j, V_j) , such that a fill rate above the prescribed value is assured. Then, the procedure $Servers\_Inventory(\tilde{F_o})$ can be described in detail as follows.

$Servers\_Inventory(\tilde{F_o})$

Let $MC(\tilde{F_o}) = OT(\tilde{F_o}) + \sum_{j \in \tilde{F_o}} (Local\_SI(0, \infty, OT(\tilde{F_o}), V_j, k_j))$

If $min\_cost > MC(\tilde{F_o})$

For $k_C = \lfloor \frac{\gamma_C}{\mu_C} \rfloor + 1$ to $\lfloor \frac{min\_cost - MC(\tilde{F_o})}{s_C} \rfloor$ do

For $V_C = V_C^{max}(min\_cost - MC(\tilde{F_o}) - k_C s_C)$ down to 0 do

$SI(k_C, V_C) = E(I_C)h_C + s_C k_C +$

$\sum_{j \in \tilde{F_o}} Local\_SI(k_C, V_C, OT(\tilde{F_o}) + E(I_C)h_C + s_C k_C, V_j, k_j)$

If $SI(k_C, V_C) + OT(\tilde{F_o}) < min\_cost$ then

$min\_cost = SI(k_C, V_C) + OT(\tilde{F_o})$

remember k_C and V_C

Return $\underset{(k_C, V_C)}{\text{minimum}}\ SI(k_C, V_C)$.

We conclude by presenting the procedure $Local\_SI(k_C, V_C, Cost\_so\_far, V_j, k_j)$, which optimizes the stock and servers at a local facility location $j \in \tilde{F_o}$. Note that, in this procedure, $(k_C, V_C, Cost\_so\_far)$ are input parameters, whereas V_j and k_j are output parameters. Recall that min_cost was a variable storing the minimum expected cost of an analysed solution. As in the case of the central facility, we can define a maximal available budget for acquiring servers and inventory at location j. As before, this budget is the amount that can be spent on servers and inventory without increasing the costs of the solution above min_cost. Deciding the inventory and the number of servers is done as follows. For every affordable (within the budget) amount of inventory, we check if, within the available budget, we can acquire the amount of servers necessary to ensure a fill rate higher then the prescribed one (α). The cheapest combination of servers and inventory is chosen among the feasible ones. More precisely, $Local\_SI(k_C, V_C, Cost\_so\_far, V_j, k_j)$ proceeds as follows.

$Local\_SI(k_C, V_C, Cost\_so\_far, V_j, k_j)$

Let $k_j^{min} = \lfloor \frac{\gamma_j}{\mu_j} \rfloor + 1$
Let $budget = min\_cost - Cost\_so\_far$
Let $min\_ser\_cost = \infty$
$V = 1$
While $Fillrate(V-1, k_j^{min}, \gamma_j, V_C, k_C) \leq \alpha$ and $k_j^{min} s_j + E(I_j)h_j \leq budget$ do
    $k = k_j^{min}$
    While $Fillrate(V, k, \gamma_j, V_C, k_C) \leq \alpha$ and $ks_j + E(I_j)h_j \leq budget$ do
        $k = k + 1$
    If $ks_j + E(I_j)h_j \leq min\{budget, min\_ser\_cost\}$ then
        $min\_ser\_cost = ks_j + E(I_j)h_j$
        $V_j = V$ and $k_j = k$
    $V = V + 1$
Return min_ser_cost

In the next section we will present some computational results obtained with the heuristic described above.

5 Computational study

This section focuses on numerical results for a number of randomly generated instances. After describing how we constructed the test instances, we discuss the results obtained by the algorithm we proposed in Section 4. For some cases we give a graph of the layout and the chosen locations of the repair shops and discuss the influence of the parameters on the obtained solutions. Since the problem was not treated in the literature before, we can not compare our results to known ones.

We compare results obtained by three different approaches. We call the procedure that checks all the possible configurations of open facilities and inventories and chooses the one with the lowest cost "Brute Force". Let "Sequential Optimization" be the procedure that first opens facilities at the locations which give minimum costs, without looking at inventories and servers, and then optimizes the number of servers and the inventories at these locations. Finally, we refer to the procedure described in Section 3 as "Local Search". In all three procedures, the central facility is replaced by the special server as done by Avsar and Zijm in [4] and described in Section 3.

We consider 7 sets of instances with 50 demand points and 15 locations where repair shops may be opened. Set I can be considered as the basic setting and the other sets as its variants. For all instances, the positions (x, y) of the demand points and the facility locations are taken randomly on $(-1, 1) \times (-1, 1)$, i.e. both x and y are $U(-1, 1)$ where $U(a, b)$ denotes the uniform distribution with parameters a and b. The central facility is located at (0,0). We set the cost of a central server $s_C = 40$, the expected central service time $1/\mu_C = 0.9$, the central inventory cost per item $h_C = 1.5$ and the fill rate $\alpha = 0.95$. The other key parameters of the system are defined as shown in Table 1.

Parameters	**Set I**	**Set II**	**Set III**	**Set IV**	**Set V**	**Set VI**	**Set VII**
Costs of opening facilities f_j	$U(30, 40)$	$U(30, 40)$	$U(30, 40)$	$U(30, 40)$	$U(30, 40)$	$U(10, 20)$	$U(10, 20)$
Costs of local servers s_j	$U(10, 20)$	$U(10, 20)$	$U(10, 20)$	$U(20, 30)$	$U(20, 30)/4$	$U(10, 20)$	$U(10, 20)$
Costs of local inventory (per item) h_j	$f_j/4$	$f_j/4$	$f_j/4$	$f_j/4$	$f_j/4$	f_j	$f_j/4$
Repair probability χ	0.95	0.95	0.95	0.99	0.90	0.95	0.95
Expected local repair time $1/\mu_j$	$U(0.1, 0.2)$	$U(0.1, 0.2)$	$U(2, 3)$	$U(0.1, 0.2)$	$U(0.1, 0.2)$	$U(0.1, 0.2)$	$U(0.1, 0.2)$
Internal transportation costs w_L	10	1	10	10	10	10	10
External transportation costs w_D	5	5	5	5	5	5	30
Demand rate λ_i	U(10,50)	U(10,50)	U(10,50)	U(10,50)	U(10,50)	U(10,50)	U(0.02,0.04)

Table 1: System parameters

First of all, these experiments were aimed to check the quality and the speed of the proposed "Local Search" heuristic. Therefore, we compare the proposed heuristic with the "Brute Force" algorithm and with the "Sequential Optimization" approach. In Table 2 we present the average and the minimal and maximal values of the relative errors of the "Local Search" heuristic and the "Sequential Optimization" procedure with respect to the "Brute Force" approach. For the instances with 15 locations, the "Local Search" heuristic was, on the average, 8.96 times faster than the "Brute Force" approach. For instances with more locations, the differences in speed are even bigger. On the other hand, the "Sequential Optimization" heuristic was, on the average, 13.84 times faster than the "Local Search" heuristic. The proposed "Local Search" heuristic requires more computational time, since

	Relative error "Local Search"	**Relative error "Sequential Optimization"**
Average	0.16%	4.24%
Maximal	9.42%	29.58 %
Minimal	0.00%	0.00 %

Table 2: Relative errors with respect to the "Brute Force" approach

for each combination of open facilities which is analysed, it optimizes the stock levels and the numbers of servers. However, as can be seen in Table 2, the experiments show that "Local Search" finds a better solution than "Sequential Optimization".

	Set I		Set II		Set III		Set IV		SetV		Set VI		Set VII	
	Heur.	Seq.	Heur.	Seq.	Heur.	Seq.	Heur.	Seq.	Heur.	Seq.	Heur.	Seq.	Heur.	Seq.
Aver.	0.01	3.23	0.00	1.84	0.00	4.42	0.00	3.82	0.03	1.68	0.00	1.94	1.22	13.15
min.	0.00	1.51	0.00	0.61	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.31	0.00	0.00
max.	0.12	5.00	0.00	3.24	0.00	7.63	0.00	13.56	0.25	6.55	0.00	3.68	9.42	29.58

Table 3: Comparison of error of different heuristics (in %)

It is interesting to compare the errors obtained for different parameter settings (Table 3). In the experiments in set VII, the big errors occur because of the low demand. That is, given the low demand, and reasonably high installation costs, only one facility is opened in the optimal solution. In our heuristics, we start with all facilities open. Apparently, while closing down the facilities, the procedure gets trapped in a local optimum. When the procedure starts with all facilities closed, "Local Search" finds the optimal solution in all the instances in set VII but "Sequential Optimization" still gets a maximal error of 21%. For other choices of the parameters, however, starting with all facilities closed, both heuristics give worse results than our original approach. The best results could be found, at the cost of computational time, by a combination of the two approaches; start the procedure with all facilities open and if in the found solution only a few facilities are open, restart the procedure with all facilities closed and take the best solution of the two approaches.

In the remaining of this section, we discuss the behavior of the solution found by "Local Search" depending on the input parameters. First of all we focus on set I, IV and V, where a decrease in the server price together with a decrease in the local repair probability, causes that more facilities are opened and that the facilities tend to be closer to the center (see Figure 4). Due to a smaller local repair probability, the transportation costs from the local repair shops to the central one play a bigger role.

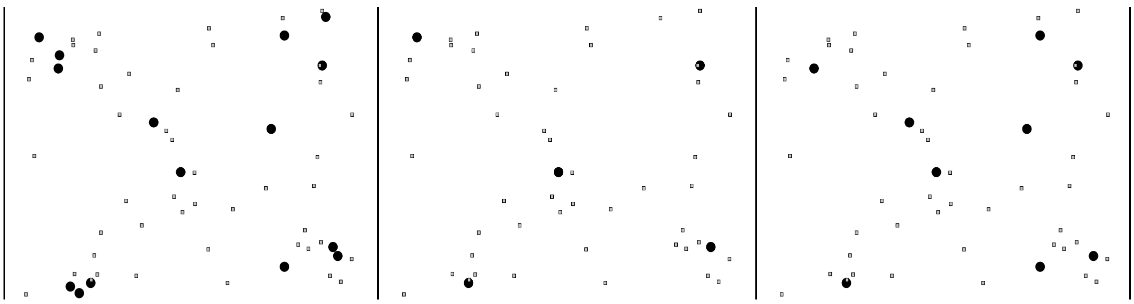

Figure 4: Example of optimal locations: possible locations(left), optimal facility locations for higher (center) and lower (right) local repair probabilities

Other factors that we have analysed are the numbers of servers (Table 4) and the inventory levels (Table 5). For each instance, we have first determined the average number of installed servers per open location and the total number of servers installed in the system. Then we have found the average, minimum and maximum of these quantities for each set of instances. Finally, we have used the same procedure for the inventory levels.

	Set I		Set II		Set III		Set IV		SetV		Set VI		Set VII	
	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total
Aver.	27.0	276.2	26.6	276.8	323.2	2939.3	30.6	222.7	23.8	328.3	22.4	271.1	1	2
min.	21.3	260	21.3	264	259.0	2892	22.6	202	20.9	241	17.9	239	1	2
max.	32.3	297	29.6	297	371	2990	42.2	245	27.6	361	26.3	300	1	2

Table 4: Comparison of average and total number of servers in the system

	Set I		Set II		Set III		Set IV		SetV		Set VI		Set VII	
	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total	aver.	total
Aver.	80.9	679.6	80.1	681.4	474.5	4283.5	78.4	546.6	70.2	714.8	68.6	675.6	2.8	2.8
min.	61.4	603	61.4	644	388.1	4164	66.0	471	45.4	460	58.6	597	2	2
max.	92.2	737	91.3	737	534.8	4398	96.6	655	80.5	779	73.5	769	3	3

Table 5: Comparison of average and total inventory levels in the system

Obviously, the increase in the service time (Set III) causes an increase in the number of servers and the stock levels. We can also see that the change of the transportation cost from the local facilities to the central (Set II) has little influence on the numbers of installed servers and the stock levels. Set IV and V indicate that the smaller the local repair probabilities and the local server costs, the lower the average amount of installed servers and the average stock levels but the higher the sums of these values. Intuitively, this is because sharing of servers is less necessary when the servers cost less and so more facilities will be opened (cf. Figure 4). The results for set VI show that even for high stock and service cost the optimal stock levels and numbers of servers remain on approximately the same level. The last set of parameter settings results in low stock levels and small numbers of servers due to the small demand. In these cases only one local facility is opened.

6 Concluding remarks

In this paper we have developed a model for a logistic problem that combines facility location with spare part management. Under specific assumptions (exponential service times, Poissonian demand), we were able to analyse the model and consequently propose a local search heuristic for finding a solution of the problem. The experimental results show that the algorithm we have designed, behaves 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 repair shop. A further research topic is to analyse the situation where the customers may be assigned to a more distant repair shop.

References

1. Aardal K. Reformulation of capacitated facility location problems: how redundant information can help. Annals of Operations Research 1998; 82: 289–308.

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

3. Arya V, Garg N, Khandekar R, Pandit V, Meyerson A, Munagala K. Local search heuristics for k-median and facility location problems. In: Proceedings of the 33rd ACM Symposium on Theory of Computing, 2001: 21-29.

4. Avsar ZM, Zijm WHM. Capacitated Two-Echelon Inventory Models for Repairable Items Systems. In: Gershwin SB et al. (Eds.): Analysis and Modeling of Manufacturing Systems. Boston: Kluwer Academic Publishers, 2002.

5. Batta R. Single server queuing-location models with rejection. Transportation Science 1988; 22: 209-216.

6. Batta R. A queuing location model with expected service time dependent queuing disciplines. European Journal of Operational Research 1989; 39: 192-205.

7. Beasley JE. Lagrangean heuristics for location problems. European Journal of Operational Research 1993; 65: 383-399.

8. Berman O, Larson R, Chiu S. Optimal server location on a network operating as a M/G/1 queue. Operations Research 1985; 12: 746-771.

9. Berman O, Larson R, Parkan C. The stochastic queue p-median location problem. Transportation Science 1987; 21: 207-216.

10. Berman O, Mandowsky R. Location-allocation on congested networks. European Journal of Operational Research 1986; 26: 238-250.

11. Christofides N, Beasley JE. Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. European Journal of Operational Research 1983; 12: 19-28.

12. Cornuejols G, Sridharan R, Thizy J. A comparison and relaxations for the capacitated plant location problem. European Journal of Operational Research 1991; 50: 280-297.

13. Davis P, Ray T. A branch and bound algorithm for the capacitated facility location problem. Naval Research Logistic Quarterly 1969; 16: 331-334.

14. Drezner Z (Ed.). Facility Location: A Survey of applications and methods. New York: Springer-Verlag, 1995.

15. Hesse Owen S, Daskin MS. Strategic facility location: a review. European Journal of Operational Research 1998; 111: 423-447.

16. Korupolu MR, Plaxton CG, Rajaraman R. Analysis of a Local Search Heuristic for Facility Location Problems. Journal of Algorithms 2000; 37(1): 146-188.

17. Kuehn AA, Hamburger MJ. A heuristic program for locating warehouses. Management Science 1963; 9: 643-666.

18. Larson RC. A hypercube queueing model for facility location and redestricting in urban emergency services. Computers and Operations Research 1974; 1: 67-95.

19. Marianov V, Serra D. Probabilistic, maximal covering location-allocation models for congested systems. Journal of Regional Science 1998; 38: 401-424.

20. Marianov V, Serra D. Hierarchial location-allocation models for congested systems. European Journal of Operational Research 2001; 135: 195-208.

21. Marianov V, Serra D. Location-allocation of multiple-server service centers with constrained queues or waiting times. Annals of Operations Research 2002; 111: 35-50.

22. Mirchandani P, Francis R (Eds). Discrete Location Theory. New York: John Wiley and Sons, Inc, 1990.

23. Muckstadt JA. A Model for a Multi-Item Multi-Echelon Multi-Indenture Inventory System. Management science 1973; 20: 472-481.

24. ReVelle C. Facility sitting and integer-friendly programming. European Journal of Operational Research 1993; 65: 147-158.

25. Ross SM. Introduction to probability models. Harcourt Academic Press, 2000.

26. Sherbrooke CC. METRIC: A Multi-Echelon Technique for Recoverable Item control. Operations Research 1968; 16: 122-141.

27. Sleptchenko AV. Integral Inventory Control in Spare Parts Networks with Capacity Restrictions. PhD thesis, University of Twente, 2002.

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

29. Wolff RW. Poisson arrivals see time averages. Operations Research 1982; 30: 223-231.