Inventory planning for spare parts networks with delivery time requirements

Inventory planning for spare parts networks with delivery time requirements

I.C. Reijnen • T. Tan* • G.J. van Houtum
i.c.reijnen@tue.nl • t.tan@tue.nl • g.j.v.houtum@tue.nl

School of Industrial Engineering, Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, Netherlands

July 10, 2009

Abstract

Motivated by real life, we introduce a new inventory model for spare parts where we explicitly take delivery time requirements into account. In this single-echelon, multi-location network, demand of a customer can be satisfied from multiple warehouses, but only if the customer can be reached from a warehouse within a time limit that is specified in the service contract of the customer. A delivery to a customer from a warehouse other than the closest one is referred to as a lateral transshipment. We develop a fast and accurate approximate algorithm to evaluate the performance of the network under given base stock levels and propose a fast and effective heuristic to set base stock levels. Numerical experiments show that planning with lateral transshipments can lead to cost savings up to 47% when compared to planning without lateral transshipments. Furthermore, we show the importance of taking lateral transshipments into account when designing a spare parts network.
Keywords: inventory, spare parts, multiple locations, lateral transshipment, delivery time constraint

1. Introduction

Consider the spare parts inventory control problem for distribution networks consisting of multiple local warehouses and a central warehouse for an Original Equipment Manufacturer (OEM) of advanced capital goods. Machines are installed at the customers of the OEM. Whenever a critical part of a machine at a customer fails, it has to be replaced by a spare part. Customers have service

*Corresponding author

contracts with the OEM to ensure that spare parts are delivered within a pre-specified time limit. To meet this time limit a certain level of spare parts is kept on stock in the local warehouses.

A customer can often be reached within the time limit from multiple warehouses. Generally the nearest warehouse is used to satisfy demand if it has stock on hand. When a demand occurs and the nearest warehouse is out of stock, another warehouse might be considered to satisfy the demand. This other warehouse should have stock on hand and it should be able to reach the customer within the time limit. In case a demand is satisfied by a regional warehouse other than the nearest warehouse, we refer to this as a lateral transshipment. If a spare part cannot be delivered from stock from one of the regional warehouses within the time limit, then an emergency shipment from the central warehouse takes place. This situation is illustrated by Figure 1. Figure 1 shows a map of Europe with 6 regional warehouses. From each of the warehouses, customers that are situated within the time limit, can be reached by that warehouse. This is illustrated by a circle drawn around each warehouse. As we can see, a customer in Lisbon can only be reached within the time limit from the warehouse in Madrid. But a customer in Nuremberg can be reached by 3 warehouses, namely Dortmund, Milan and Budapest. Dortmund is the closest warehouse for a customer in Nuremberg. So, when a customer in Nuremberg requires a spare part, Dortmund is first asked to deliver a spare part. In case Dortmund delivers the spare part, this delivery is called a regular shipment. When Dortmund cannot deliver the spare part, then Milan is requested to send the spare part, and if Milan can also not deliver, then Budapest is considered. If Milan or Budapest delivers the spare part, this is called a lateral transshipment. If none of the three warehouses can deliver, then an emergency shipment from the central warehouse satisfies the demand.

In this paper we focus on the inventory planning of the regional warehouses, and therefore assume that the central warehouse has ample stock, but can never reach a customer within the time limit. We note that a customer that can be reached within the time limit by the central warehouse will always be delivered on time. Therefore, such customers can be left outside the scope of our model. The above model is used in the rest of our paper to demonstrate our model. For the sake of clarity, we assume that the central warehouse is located in another continent in this situation.

The model we consider in this paper is motivated by the spare parts network we encountered at an OEM of high-tech equipment. The OEM wants to determine stock levels such that the sum of transportation, emergency shipment, and inventory holding costs is minimized under a given service level constraint. In this case, the service measure is the fraction of demands that is satisfied within the time constraint, which we refer to as the time-based fill rate. One typically requires that this fraction is at least equal to 90% and is set per item.

The contribution of this paper is as follows:

- *We introduce a new model for spare parts networks with delivery time constraints.* The

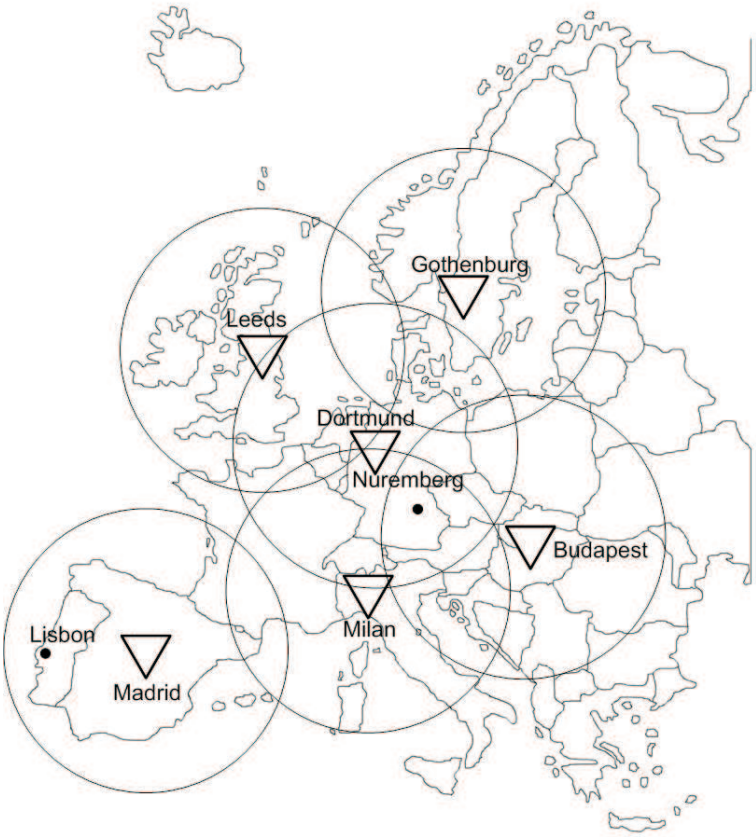


Figure 1: Spare parts network Europe

service contract of a customer specifies a time limit on the delivery time of a spare part in case of failure. The model considers for each customer whether a delivery within this time limit is possible from each of the warehouses, which we refer to as the customer being 'reachable'. A transshipment from a warehouse to a customer is possible only if the customer is reachable from that warehouse. This results per customer in an ordered list of warehouses for the delivery of a spare part. For example, customers in the area of Nuremberg have an ordered list 'Dortmund, Milan, Budapest', while customers situated a bit more south can have an ordered list of 'Milan, Budapest, Dortmund'. Customers in the area of Lisbon have an ordered list consisting of Madrid only. Hence, the ordered list can differ per group of customers and can differ in length. To the best of our knowledge, such a flexible model for lateral transshipments has not been considered before.

- *We develop a fast and accurate approximate evaluation algorithm that is applicable for real-life instances.* This algorithm evaluates the performance per Stock Keeping Unit (SKU) under given stock levels at the warehouses to determine the total costs and the service levels.

Exact evaluation is possible via Markovian analysis, but can handle only very small instances. The approximate evaluation algorithm can evaluate real-life instances within 0.001 seconds. By simulation we show that the algorithm estimates the time-based fill rates with an error no more than 2%.

- *We provide an efficient and effective heuristic solution based on the approximate evaluation algorithm to set the base stock levels.* The algorithm is based on a greedy procedure and for small instances we show that it finds solutions with corresponding costs that are on average within 2 % of the optimal costs.

- *We provide new managerial insights.* By numerical comparison we show the importance of taking lateral transshipments into account in the planning phase. We conclude that the savings per SKU can go up to 47 %. Furthermore, we show that the total costs of inventory and transportation costs decrease in the number of warehouses. This implies that when designing a spare parts network under delivery time requirements, a larger number of warehouses is likely to be optimal, instead of the more traditional model where the number of warehouses is minimized.

The network model we consider is motivated by a time limit on the demand fulfillment. This time limit on the demand fulfillment results in a general structure for the use of lateral transshipments. In this structure, inventory is shared only partially among the warehouses, which is often referred to as partial pooling. Different forms of partial pooling have been studied before in the literature. We distinguish two types of partial pooling: (i) pooling is limited by the network structure, and (ii) pooling is limited by holding back inventory. Pooling that is limited by holding back inventory is mainly considered in a game theoretical setting [18] or when lead times are non-negligible [13]. Since this paper is concerned with pooling limited by the network structure, we focus on papers in that area.

The most common form of this type of partial pooling is to use pooling groups, where the local warehouses are divided into groups and complete pooling is applied within a pooling group while no lateral transshipments are allowed between groups. Lee [10] considers a two-echelon, single-item, multi-location, continuous review model with a base stock policy. The model is concerned with a network where the regional warehouses are grouped into pooling groups. In case a regional warehouse is out of stock, a lateral transshipment can take place from a randomly chosen other warehouse with on-hand stock that is within the pooling group. If all warehouses in the pooling group are out of stock, then the request is sent to the central warehouse. In case the central warehouse is also out of stock, then the demand is backlogged. Axsäter [2] considers the same model. An important contribution of Axsäter [2] is that he provides a method to estimate the demand rates observed by each regional warehouse, taking the lateral transshipments into account.

If the geographical area where the customers are situated is small compared to the time limit, so that each customer can be reached by each warehouse, then our situation is equivalent to the special case of complete pooling. Complete pooling has been studied by e.g. Kukreja et al. [6], Alfredsson and Verrijdt [1], and Wong et al. [16]. Kukreja et al. [6] use a fixed order for the choice of the warehouse to provide a lateral transshipment, which could be based on distances as in our case. Alfredsson and Verrijdt [1] choose randomly among the warehouses with available stock. Wong et al. [16] consider complete pooling and allow for delayed lateral transshipments. Delayed lateral transshipments occur when a part is backordered at a warehouse while another warehouse receives a replenishment.

Another structure for lateral transshipments is, for example, unidirectional lateral transshipments as considered by Axsäter [3], Liu and Lee [11], and Olsson [12] in a single-echelon, multi-location system. Unidirectional lateral transshipments are relevant when shortage costs differ strongly per warehouse, and are very similar to demand substitution. Kranenburg and van Houtum [5] study a partial pooling structure for lateral transshipments where complete pooling is applied among main local warehouses and regular local warehouses can only receive lateral transshipments. All the models mentioned here can be formulated as special cases of our model. We refer the reader to Wong *et al.* [17] for a more extensive review on lateral transshipments. To the best of our knowledge, a structure for lateral transshipments that is based on time limits has not been studied before.

The time-based fill rate as a service measure has been studied before by Kutanoglu [8], Kutanoglu and Mahajan [9] and Caggiano et al. [4]. In the models of Kutanoglu [8] and Kutanoglu and Mahajan [9] complete pooling is applied, and base stock levels are optimized under time-based service level constraints. The main difference between their models and our model is the partial pooling structure that we use for lateral transshipments. Caggiano et al. [4] study time-based fill rates in a distribution setting where lateral transshipments are not considered at all.

The approximate evaluation algorithm we develop, follows the line of reasoning of Axsäter [2]. Axsäter approximates the demand rates observed by each warehouse by assuming that all demand streams are Poisson. Based on the resulting set of equations Axsäter [2] provides an iterative algorithm to obtain steady state probabilities. Similar algorithms for different models have been developed by Alfredsson and Verrijdt [1], Kukreja et al [6], Kutangolu [8], and Kranenburg and van Houtum [5]. The approximate evaluation algorithm we develop provides a different set of equations for the iterative algorithm such that it is applicable for a network with delivery time requirements.

This paper is organized as follows. In Section 2 we describe our model. In Section 3 we describe how the model can be evaluated in an exact way and show how the optimal base stock levels can be found. In Section 4.1 we introduce our approximate evaluation method. Next, in Section 4.2, we describe the greedy algorithm to determine order-up to levels. In Section 5 we present

computational results on the sensitivity of the model to the choice of exponential or deterministic replenishment lead times, the performance of the evaluation method, and the performance of the greedy algorithm. In Section 6, based on case data of the OEM of high-tech equipment, we provide managerial insights. In Section 7, we propose some possible extensions. We conclude in Section 8.

2. Model description

The installed machines at a customer consist of multiple critical parts which can be replaced by a spare part in case of failure. For each spare part a stock keeping unit (SKU) is defined for which stock is kept at several warehouses. The customers have service contracts with the OEM where the OEM promises to deliver a demanded spare part within a certain time limit. In case all warehouses that can reach a customer are out of stock, the OEM sends an emergency shipment against extra costs. The OEM works with a constraint on the time-based fill rate per SKU, the fraction of demand that is fulfilled within the time limit of a customer. This time-based fill rate is set to be at least γ_0 by the management of the OEM and reflects both the extra costs that are made for an emergency shipment and the loss of goodwill at the customer. Since the constraint on the time-based fill rate is set per SKU, we can analyze the system with a single item model and we consider one specific critical component. Let N be the set of all (groups of) customers. Customers are spread over a geographical area, for example Europe. In case customers are located very close to each other, then we can decide to combine the customers into a group of customers to reduce the number of customers in the model. So a customer in the model might represent a group of customers in reality. The demand for the item at customer n is assumed to follow a Poisson process with constant rate μ_n.

Let $J = \{1, 2, .., |J|\}$ be the set of regional warehouses. The central warehouse is assumed to have ample stock. Each regional warehouse $j \in J$ uses a base stock policy for the replenishment of the SKUs. The replenishment lead time for warehouse $j \in J$ is denoted by t_j^{reg} and the base stock level of the SKU by S_j, where $\mathbf{S}$ denotes the vector of all base stock levels. The replenishment lead times are assumed to be i.i.d. and the distribution is assumed to be exponential with mean t_j^{reg}. In practice we observed that the lead time follows a deterministic distribution. However, we assume exponential replenishment lead times to facilitate exact evaluation via Markov processes. In Section 5, we show that the performance of the model is not sensitive for the choice of exponential lead times or deterministic lead times, and thus all our insights for the case with exponential lead times hold also under deterministic lead times.

For each customer n we define an array υ_n of warehouses that can reach customer n within the time limit, where $\upsilon_n(i)$ denotes the i^{th} warehouse in the array ($i = 1, 2..., p_n$), with $p_n \geq 0$ the length of $\upsilon_{\mathbf{n}}$. The array can be ordered such that you have increasing transportation times, but

other orderings are possible as well. The array can also be empty. When customer n needs a spare part it will first submit its request to warehouse $\upsilon_n(1)$. In case that warehouse is out of stock, it will turn to warehouse next in the array ($\upsilon_n(2)$), and so on until the demand is satisfied, or there are no more warehouses in the array. A shipment from warehouse $\upsilon_n(i)$ with $i \geq 2$ is called a lateral transshipment. In case the demand is not satisfied an emergency shipment will take place.

The costs we deal with are (lateral) transshipment, emergency shipment and inventory holding costs. We assume that replenishment costs are equal for all warehouses. Then for each demand we have at least the costs for a regular replenishment. Emergency shipment costs are denoted by $c_{n,0}$ and we define them as the extra costs for a fast shipment from the central warehouse to customer n compared to a regular replenishment to a warehouse and delivery to the customer. Under these assumptions, the yearly regular replenishment costs have to be accounted for all demands. They then form a constant factor, and therefore we can and do exclude them from our cost accounting. Transshipment costs $c_{n,j}$ are the costs incurred to transport a spare part from warehouse j to customer n. Inventory holding costs h_j are incurred per time unit for each item on stock at warehouse j and in the replenishment pipeline per time unit. Since a base stock policy is applied, the sum of the items on stock and in the pipeline for warehouse j is always equal to S_j, and thus the inventory holding costs are equal to $\sum_{j\in J} h_j S_j$.

With respect to the demand fulfillment we introduce the following notation:

- $\alpha_{n,j}$, for the fraction of demand of customer n that is fulfilled by warehouse j;
- $\alpha_n = \sum_{j\in J} \alpha_{n,j}$, for the fraction of demand of customer n that is fulfilled by the regional warehouses;
- θ_n for the fraction of demand of customer n that is satisfied by an emergency shipment.

Note that, since each demand is fulfilled from either a warehouse $j \in J$ or an emergency shipment, at each customer $n \in N$, it holds that $\theta_n + \alpha_n = 1$. The expected total costs incurred per time unit for the shipments equals $\sum_{n\in N} \left(\sum_{j\in J} \mu_n \alpha_{n,j} c_{n,j}\right)$. The expected total costs incurred per time unit for emergency shipments equals $\sum_{n\in N} \mu_n \theta_n c_{n,0}$. Therefore the expected total costs per time unit $C(\mathbf{S})$ is

$$C(\mathbf{S}) = \sum_{j\in J} S_j h_j + \sum_{n\in N} \left(\mu_n \theta_n c_{n,0} + \sum_{j\in J} \mu_n \alpha_{n,j} c_{n,j} \right). \tag{1}$$

The fraction of demand that is delivered within the time limit for the SKU is equal to:

$$\gamma(\mathbf{S}) = \frac{\sum_{n\in N} \mu_n \alpha_n}{\sum_{n\in N} \mu_n}. \tag{2}$$

We refer to $\gamma(\mathbf{S})$ as the time-based fill rate. To fulfill the time-based fill rate constraint, the fraction γ_0 of the total demand should be fulfilled by a regional warehouse. This can be expressed by the inequality $\gamma(\mathbf{S}) \geq \gamma_0$.

The objective is to minimize the total expected costs per time unit under the time-based fill rate constraint. This problem can be formulated mathematically as follows:

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ s.t. \quad & \gamma(\mathbf{S}) \geq \gamma_0. \end{aligned}$$

3. Exact evaluation and optimization

For small instances of the model, we can evaluate the total costs and the time-based fill rate exactly. We can use this evaluation to optimize the base stock levels, but only for very small instances. In this section we describe the exact evaluation and optimization.

3.1 Exact evaluation

The model described in Section 2 can be evaluated exactly with Markovian analysis for a given choice of base stock levels. In this subsection we describe how this can be done. In the evaluation we determine values for $\alpha_{n,j}$, $n \in N$, $j \in J$ and θ_n, $n \in N$. From these values we can immediately determine the costs and the time-based fill rate from Equations 1 and 2.

For the exact evaluation we define a $|J|$-dimensional Markov process, with $\mathbf{x} = (x_1, x_2, ..., x_{|J|})$, where x_j, $j \in J$ denotes the on-hand inventory at local warehouse j, with $0 \leq x_j \leq S_j$, $j \in J$. Furthermore, define e_j as the row vector with the j-th element equal to 1 and all other elements equal to 0.

When the system is in state x, then two types of events can occur. Either a replenishment arrives at a warehouse $j \in J$, or a customer generates a demand. By considering all events for each state x, we can identify all transitions.

When the system is in state x with $x_j < S_j$ for a regional warehouse $j \in J$, then a replenishment can arrive at warehouse j. When a replenishment arrives, we observe a state transition $x \to x + e_j$. In state x there are $S_j - x_j$ items in replenishment, so the transition rate for the state transition $x \to x + e_j$ is $(S_j - x_j)/t_j^{reg}$.

When the system is in state x and customer n generates a demand, then either a reachable regional warehouse will fulfill the demand, or an emergency shipment takes place. In case there is at least one reachable warehouse with stock on hand, then let warehouse $j \in J$ be the first warehouse in the array υ_n with stock on hand, so $x_j > 0$. In this case, the state transition is $x \to x - e_j$,

and the transition rate is μ_n. In case none of the reachable warehouses has stock on hand, then an emergency shipment will take place and no state transition will occur.

The number of states equals $\prod_{j=1}^{|J|}(S_j+1)$ and therefore computation time is exponential in the number of local warehouses. This implies that exact evaluation can only be done for a limited number of regional warehouses.

When all transitions are determined, we can can find the steady-state probabilities of the Markov process by the power method. The power method uses successive approximation and theoretically converges to the exact steady-state probabilities. However, we will stop when steady-state probabilities do not change more then ϵ, with ϵ small, to avoid extreme long computation times.

Let $\pi(x)$ be the steady-state probability of state x.From the steady-state probabilities we can determine the values for $\alpha_{n,j}$ and θ_n as follows:

$$\alpha_{n,\upsilon_n(j)} = \sum_{x|x_{\upsilon_n(k)}=0 \ k<j, \ x_{\upsilon_n(j)}>0} \pi(x)$$

$$\theta_n = \sum_{x|x_i=0 \forall i=1\ldots|\upsilon_n|} \pi(x).$$

3.2 Optimization

The base stock levels can be optimized by "smart enumeration". We start the enumeration with a total base stock level $S_T = \sum_{j\in J} S_j$ of 0 and then we increment S_T. In order to avoid evaluation of all possible combinations, we introduce two upper bounds on the time-based fill rate and two lower bounds on the total costs. The first upper bound on the time-based fill rate is $\gamma_1^u(S_T)$. This upper bound is obtained by assuming that all customers are reachable by all warehouses and therefore only requires the information of the total stock $S_T = \sum_{j\in J} S_j$ at the warehouses; thus the precise location of the stock is not important. Consider a single demand stream with demand rate $\mu = \sum_{n\in N} \mu_n$ and a single virtual warehouse with replenishment lead time $t_{max} = \max_{j\in J} t_j^{reg}$. To determine the fill rate of this warehouse we can use known results from the Erlang loss system (also denoted as an M—G—c—c queueing system), since the steady state behavior of the stock is identical to the steady state behavior of idle servers in an Erlang loss system with arrival rate μ, S_T servers, and the mean replenishment lead time t_{max} as mean service time. Thus, we can determine the probability of having stock on hand via the Erlang loss probability. The Erlang loss probability $L(c,\rho)$, where c represents the number of servers and ρ the offered load (obtained as the product of the arrival rate and the mean service time), is given by (see e.g. Tijms [14]).

$$L(c,\rho) = \frac{\rho^c/c!}{\sum_{x=0}^{c} \rho^x/x!} \tag{3}$$

The upper bound $\gamma_1^u(S_T)$ is the fill rate of the virtual warehouse and is given by:

$$\gamma_1^u(S_T) = 1 - L(S_T, t_{max}\mu).$$

The second upper bound on the time-based fill rate is $\gamma_2^u(\mathbf{S})$, and for this upper bound the exact distribution of the stock over the warehouses is important. The upper bound is obtained by first isolating the customers that cannot be reached by any of the warehouses with $S_j > 0$. For those customers we know that $\alpha_n = 0$. For the other customers we determine an upper bound $\tilde{\alpha_n}$ on α_n by assuming that they are reachable by all warehouses with a replenishment lead time of $t_{max,2} = \max_{j \in J | S_j > 0} t_j^{reg}$. Let $\tilde{N}$ be the set of these customers and define $\tilde{\mu} = \sum_{n \in \tilde{N}} \mu_n$. Then, $\tilde{\alpha}_n = 1 - L(S_T, t_{max,2}\tilde{\mu})\ \forall n \in \tilde{N}$ and $\tilde{\alpha}_n = 0\ \forall n \in N \setminus \tilde{N}$, yielding $\gamma_2^u(\mathbf{S}) = \sum_{n \in N} \mu_n \tilde{\alpha}_n / \sum_{n \in N} \mu_n$.

The first straightforward lower bound on the total costs is $C_1^l(S_T) = S_T * min_{j \in J} h_j$. Note that $C_1^l(S_T)$ only depends on S_T and is increasing in S_T. The second lower bound on the total costs is based on $\gamma_2^u(\mathbf{S})$, and defined as $C_2^l(\mathbf{S}) = \sum_{j \in J} h_j S_j + \sum_{n \in N \setminus \tilde{N}} \mu_n c_{n,0} + (1 - \gamma_2^u(\mathbf{S}))\tilde{\mu} \min_{n \in \tilde{N}} c_{n,0}$.

The smart enumeration procedure then works as follows as described in Algorithm 1.

Smart enumeration

Initialization

Determine the lowest S_T such for which $\gamma_1^u(S_T) \geq \gamma_0$,

$C_{best} = +\infty$.

Repeat while $C_1^l(S_T) < C_{best}$

For all solutions $\mathbf{S}$ with a total stock S_T:

If $\gamma_2^u(\mathbf{S}) \geq \gamma_0$ and $C_2^l(\mathbf{S}) < C_{best}$ then

evaluate the solution with the exact procedure of Section 3.1.

If $C(\mathbf{S}) < C_{best}$ and $\gamma(\mathbf{S}) \geq \gamma_0$ then $C_{best} = C(\mathbf{S})$ and $\mathbf{S}_{best} = \mathbf{S}$.

$S_T = S_T + 1$.

End

4. Approximate analysis

For large (real-life) instances, exact evaluation and optimization is not possible because of computational complexity. In this section we propose an approximate evaluation algorithm for the evaluation of the network under given base stock levels and a heuristic to set base stock levels.

4.1 Approximate evaluation

In this subsection we propose an approximate evaluation algorithm for the evaluation of the network under given base stock levels.

In the approximate evaluation algorithm we are interested in approximating values for $\alpha_{n,j}$ and θ_n under a given policy $\mathbf{S}$. The costs $C(\mathbf{S})$ and the time-based fill rate $\gamma(\mathbf{S})$ then follow from Equations (1) and (2).

The method is based on two approximate steps: (i) overflow demand streams are assumed to be Poisson distributed, and (ii) stock levels at the warehouses are assumed to be independent of each other, such that the warehouses can be analyzed as single stock points. For the algorithm we introduce the following notation.

- $M_{n,j}$, for the demand rate that warehouses j faces from customer n;
- $M_j = \sum_{n \in N} M_{n,j}$, for the total demand warehouses j faces;
- β_j, for the fraction of the total demand rate that warehouse j can deliver from stock.

For the first step of the algorithm we assume that the fill rates of the warehouses are known, and we estimate the total demand a warehouse faces. We note that the demand at warehouse j from customer n includes requests for lateral transshipments. The same request might go to multiple warehouses until one has stock on hand, so in general $\sum_{j \in J} M_{n,j} \geq \mu_n$. For warehouse $j = \upsilon_n(1)$ the total demand warehouse j faces from customer n, $M_{n,j}$, is precisely the total demand customer n generates: $M_{n,j} = \mu_n$. Suppose we know the fill rates β_j, $j \in J$, of the regional warehouses. If we assume that the stock levels at the local warehouses are independent, then we know that the demand rate warehouse $\tilde{j} = \upsilon_n(2)$ faces from customer n is on average $(1 - \beta_{\tilde{j}})M_{n,j}$. Or more general, for $2 \leq i \leq |\upsilon_n|$:

$$M_{n,\upsilon_n(i)} = (1 - \beta_{\upsilon_n(i-1)})M_{n,\upsilon_n(i-1)} \quad 2 \leq i \leq p_n. \tag{4}$$

This is illustrated in Figure 2. Warehouse 1 faces Poisson demand from a certain customer with rate $M_{n,\upsilon_n(1)} = \mu$. If we know that warehouse 1 has a fill rate of β_1, then the demand that warehouse 2 faces from this customer is on average $M_{n,\upsilon_n(2)} = (1 - \beta_1)M_{n,\upsilon_n(1)}$. Warehouse 2 can only fulfill part of this demand, namely β_2, so warehouse 3 faces $M_{n,\upsilon_n(3)} = (1-\beta_2)M_{n,\upsilon_n(2)} = (1-\beta_1)(1-\beta_2)\mu$.

If we assume that the demand warehouse j faces from customer n is a Poisson process with rate $M_{n,j}$, then the total demand warehouse j faces is a Poisson process with rate M_j. This assumption does not hold in general, but is reasonable for high fill rates. This follows from the observation that the probability of having more than 1 demand during a stock-out period is rather small.

For the second step of the algorithm we assume that the demand at a warehouse is known and estimate the fill rates. To determine the fill rates of the warehouses we can use known results from the Erlang loss system, as in Section 3.2. The fraction β_j corresponds to the probability of having stock on hand at warehouse j. Thus, we can determine $\beta_j(S_j)$ by

$$\beta_j(S_j) = 1 - L(S_j, t_j^{reg} M_j)$$

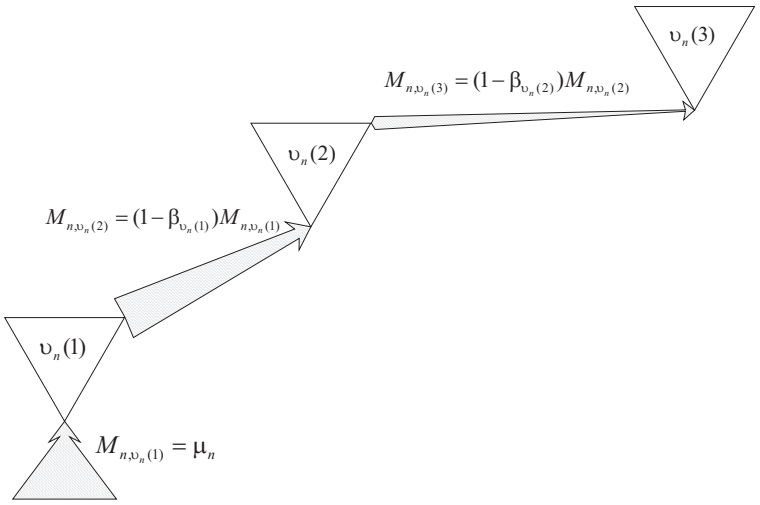


Figure 2: Illustration of overflow demand

The fraction of demand of customer n satisfied by warehouse $j \in J$ now follows from $\beta_j(S_j)$ and $M_{n,j}$.

$$\alpha_{n,j} := \frac{\beta_j M_{n,j}}{\mu_n}. \tag{5}$$

$$\theta_n = 1 - \sum_{i=1}^{p_n} \alpha_{n,\upsilon_n(i)}. \tag{6}$$

We use an iterative procedure to determine M_j, $M_{n,j}$, and $\beta_j(\mathbf{S})$ from which we derive $\alpha_{n,j}(\mathbf{S})$ and $\theta_n(\mathbf{S})$. Initially, we assume that no lateral transshipments take place, therefore we can set $M_{n,\upsilon_n(1)} = \mu_n$, and $M_{n,\upsilon_n(i)} = 0$ for $2 \leq i \leq |\upsilon_n|$. From these values we can determine $M_j = \sum_{n \in N} M_{n,j}$ and $\beta_j((S) = 1 - L(S_j, t_j^{reg} M_j)$. Given the fill rates $\beta_j(S_j)$ for the warehouses, we can obtain a better estimate for the demand flows $\widehat{M}_{n,j}$ from Equation 4. We iterate until M_j does not change more than ϵ, with ϵ small. We observed that this iterative procedure converges in all tested cases, but there is no formal proof for the convergence. After the iterative procedure, we determine from the demand flows $M_{n,j}$ the values for $\alpha_{n,j}(\mathbf{S})$ and $\theta_n(\mathbf{S})$.

The approximate evaluation algorithm can be formally described as follows:

Approximate Evaluation Algorithm

Initialization

$\forall\ j \in J,\ \beta_j := 1 - L(S_j, t_j^{reg} \sum_{n \in N | \upsilon_n(1)=j} \mu_n)$.

$\forall\ n \in N,\ M_{n,\upsilon_n(1)} := \mu_n$.

$\forall\ n \in N,\ j \neq \upsilon_n(1) \in J,\ M_{n,j} := 0$.

Repeat until M_j does not change more than ϵ for each $j \in J$:

Step 1 $\forall\ n \in N$ and for $2 \leq i \leq p_n$: $M_{n,\upsilon_n(i)} := \left(1 - \beta_{\upsilon_n(i-1)}\right) M_{n,\upsilon_n(i-1)}$.

Step 2 $\forall\ j \in J\ M_j := \sum_{n \in N} M_{n,j}$, $\beta_j := 1 - L(S_j, t_j^{reg} M_j)$

Finalization $\forall\ n \in N$ and $\forall\ j \in J$, $\alpha_{n,j}(S) := \frac{\beta_j(S) M_{n,j}}{\mu_n}$.

$\forall\ n \in N$, $\theta_n = 1 - \sum_{i=1}^{p_n} \alpha_{n,\upsilon_n(i)}$.

End

4.2 Determining Base Stock Levels

In the previous section, we have described an approximate method for the evaluation of a given policy, i.e., a choice for all base stock levels S_j, $j \in J$. We propose a heuristic solution to determine base stock levels for this problem.

Kranenburg and van Houtum[5] and Wong et al. [15] showed that for their problem and approximate evaluation algorithms greedy algorithms perform very well. Since our problem is closely related to theirs, we suggest a greedy algorithm as well.

The greedy algorithm consists of three phases, an initialization phase, a costs phase, and a time-based fill rate phase. In the initialization phase all inventory levels S_j, $j \in J$ are set to 0. In the costs phase, the greedy algorithm places inventory one by one such that the system costs are minimized each time an extra item is put on stock. It will continuously place an extra unit of inventory until no cost reduction can be achieved in this way. This is done by determining the costs difference for each warehouse when an extra unit of stock is placed. Therefore we define $\Delta C(\mathbf{S}, j) = C(\mathbf{S}+e_j) - C(\mathbf{S})$. In case the resulting stock levels are such that the $\gamma(\mathbf{S}) \geq \gamma_0$, then the algorithm terminates. Otherwise we continue with the last phase. In this phase the greedy algorithm continues with placing stock at the warehouses where the biggest improvement in service level is achieved compared to the extra total costs. Therefore we define the ratio $R(\mathbf{S}, j) = \frac{\gamma(\mathbf{S}+e_j) - \gamma(\mathbf{S})}{\Delta C(\mathbf{S}, j)}$. As long as $\gamma(\mathbf{S}) < \gamma_0$, we have not yet met the time-based fill rate constraint and we will place an extra unit of stock at the warehouse with the smallest ratio $R(\mathbf{S}, j)$. In the following, we formally state the algorithm.

Greedy Algorithm

Initialization

Set $S_j := 0$, $j \in J$

Calculate $\Delta C(\mathbf{S}, j)$, $j \in J$.

Repeat while $\min\{\Delta C(\mathbf{S}, j)\} \leq 0$:

1. Find $\hat{j} = argmin_{j \in J}\{\Delta C(\mathbf{S}, j))\}$.
2. Set $S_{\hat{j}} := S_{\hat{j}} + 1$;

3. Calculate $\Delta C(\mathbf{S}, j)$, $j \in J$.

Calculate $R(\mathbf{S}, j)$, $j \in J$.
Repeat while $\gamma(\mathbf{S}) < \gamma_0$:

1. Find $\hat{j} = argmin_{j \in J}\{R(\mathbf{S}, \hat{j})\}$.
2. Set $S_{\hat{j}} := S_{\hat{j}} + 1$;
3. Calculate $R(\mathbf{S}, j)$, $j \in J$.

End

In the greedy algorithm it is possible to use either the exact evaluation or the approximate evaluation procedure to determine $\Delta C(\mathbf{S}, j)$ and $R(\mathbf{S}, j)$. However, since computational times of the exact evaluation are too large for practical applications, we use the approximate evaluation procedure in the greedy heuristic throughout the rest of this paper.

5. Computational results

We have three objectives in conducting numerical experiments. First we show that the model is rather insensitive to the assumption of exponential replenishment lead times. Second, we test the performance of the evaluation algorithm. Third, we test the performance of the greedy algorithm by comparing the results to the optimal solution.

5.1 Test bed

To address these three objectives, we use a test bed that is based on real-life data of an OEM of high-tech equipment. The OEM currently has to plan inventory for 30,000 SKUs from which we selected 20 based on their price, demand rates, geographical spread, such that we obtained a representative selection of all SKUs. We clustered the customers into groups based on their geographical locations, such that each customer group can be seen as a single customer. The clustering of customers resulted in 400 customers. The selected SKUs and their main characteristics are shown in Table 1, where $\overline{\mu} = \frac{1}{n}\sum_{n \in N} \mu_n$, and the set of customers N only contains customers with $\mu_n > 0$ ($|N|$ varies per SKU). All prices are normalized such that the cheapest SKU is 1.00 unit.

The costs parameters are based on the parameters used by the OEM. For the shipment costs we distinguish three distance categories. For each category a different fee f_i, $i = 1, 2, 3$, is charged per kilogram, where the minimum weight is rounded up to 2 kg. The fee f_1 is applied for shipments up to a given threshold distance d_1, f_2 is applied for shipments over the interval $[d_1, d_2]$ and f_3 is applied

for shipments over a distance larger than d_2; it holds that $0 < f_1 < f_2 < f_3$ and $0 < d_1 < d_2$. The values for the fees and the threshold distances are given by the Logistic Service Provider as follows: $f_1 = 0.79$, $f_2 = 0.99$, $f_3 = 1.04$, $d_1 = 200\ km$, and $d_2 = 400\ km$. Let w represent the weight of an SKU. Let $d_{n,j}$ represent the distance from customer n to a warehouse j. The fee $f(n, j)$ that is paid for the distance $d_{n,j}$ is f_1 if $d_{n,j} \leq d_1$, f_2 if $d_1 < d_{n,j} \leq d_2$, and f_3 if $d_{n,j} > d_2$. The costs are then defined as $c_{n,\upsilon_n(1)} = f(n, \upsilon_n(1)) \max\{2, w\}$. For a lateral transshipment the costs are multiplied with a factor of 1.2, since this is not the standard procedure and requires some extra labor. The costs for a lateral transshipment are hence defined as $c_{n,\upsilon_n(j)} = 1.2 f(n, \upsilon_n(j)) \max\{2, w\}$, $1 < j \leq p_n$. An emergency shipment always comes from a distance that falls into category 3, and since the shipment has to be fast, the costs are higher than for a normal shipment, which is translated into an extra factor of 2.5. The costs for an emergency shipment are taken as $c_{n,0} = 2.5 f_3 \max\{2, w\}$. The inventory holding cost per year is estimated by the OEM as 20% of the price of an SKU.

nr.	price	w (kg)	$\overline{\mu}$	$\lvert N \rvert$
1	1.00	0.01	13.64	193
2	17.59	0.80	15.48	156
3	25.22	0.92	8.84	120
4	26.50	1.05	6.43	119
5	26.79	0.45	9.45	111
6	31.38	0.55	12.78	143
7	33.58	1.39	7.08	106
8	38.44	0.50	11.28	121
9	39.33	1.98	10.04	109
10	44.99	0.91	5.88	88
11	76.55	2.00	2.48	42
12	86.01	0.30	3.68	73
13	101.71	0.55	5.02	65
14	133.80	0.54	4.83	65
15	165.61	8.40	1.60	42
16	169.35	1.75	2.36	44
17	177.65	0.15	3.00	34
18	209.48	1.75	1.07	27
19	230.93	0.20	2.71	35
20	264.11	4.50	1.09	22

Table 1: Item characteristics

For the network we tested 5 choices of number (6, 8, 10, 12, 14) and locations of local warehouses. These numbers are based on the fact that at least 6 local warehouses are required to cover the area of the customers sufficiently and 14 is a rather high number of warehouses for the OEM. The choice

of the locations of the warehouses is such that most demand from the customers is covered by at least one warehouse and for the areas with high demand, the travel times are small.

All tests are performed with 3 different choices of base stock levels resulting in a $\gamma(\mathbf{S})$ of approximately 0.8, 0.9 and 0.95. In total we have a testbed of $20 * 5 * 3 = 300$ instances.

5.2 Sensitivity with respect to the lead time distribution

For our model we assumed that the lead times are exponentially distributed, though in practice we observed approximately constant replenishment lead times. To test the sensitivity of the performance of the model to this assumption, we simulated $\gamma(\mathbf{S})$ with both exponential ($\gamma_{exp}(\mathbf{S})$) and deterministic lead times ($\gamma_{det}(\mathbf{S})$). We simulated the difference between the model with exponential lead times and the model with deterministic lead times for all 300 instances. The simulation results are summarized in Table 2. The table shows the average difference $\overline{\Delta} = \overline{|\gamma_{exp}(\mathbf{S}) - \gamma_{det}(\mathbf{S})|}$ averaged over all SKUs, and $\Delta_{\max} = \max(|\gamma_{exp}(\mathbf{S}) - \gamma_{det}(\mathbf{S})|)$ over all SKUs. From the simulation results we can conclude that, under our form of partial pooling, the performance of the model is rather insensitive to the choice of exponential or deterministic lead times especially under higher service targets. Alfredsson and Verrijdt [1] found a similar level of insensitivity for a complete pooling situation. Hence, our results extend their findings to partial pooling.

	$\gamma(\mathbf{S}) \sim 0.8$		$\gamma(\mathbf{S}) \sim 0.9$		$\gamma(\mathbf{S}) \sim 0.95$	
scenario	$\overline{\Delta}$	$\Delta_{\max}$	$\overline{\Delta}$	$\Delta_{\max}$	$\overline{\Delta}$	$\Delta_{\max}$
6 warehouses	0.0006	0.0025	0.0004	0.0018	0.0003	0.0011
8 warehouses	0.0006	0.0025	0.0004	0.0019	0.0003	0.0016
10 warehouses	0.0006	0.0021	0.0005	0.0021	0.0003	0.0014
12 warehouses	0.0006	0.0019	0.0004	0.0019	0.0003	0.0013
14 warehouses	0.0005	0.0021	0.0005	0.0018	0.0003	0.0015

Table 2: Sensitivity of the lead time distribution

5.3 Performance of the evaluation algorithm

The performance of the evaluation algorithm is tested via simulation, where we compare the evaluated values for $\gamma(\mathbf{S}) = \gamma_{alg}(\mathbf{S})$ to the simulated values for $\gamma(\mathbf{S}) = \gamma_{sim}(\mathbf{S})$ for all 300 instances.

The results are presented in Table 3, where they are summarized over the SKUs. The table shows the average $\overline{\Delta^e} = \overline{\gamma_{alg}(\mathbf{S}) - \gamma_{sim}(\mathbf{S})}$ averaged over all SKUs and the maximum value $\Delta^e_{\max} = \gamma_{alg}(\mathbf{S}) - \gamma_{sim}(\mathbf{S})$ over all SKUs. From the results we observed that $\gamma_{alg}(\mathbf{S}) > \gamma_{sim}(\mathbf{S})$ for all tested cases, so the algorithm always overestimates the aggregate fill rate. This can be explained by the algorithm being too optimistic by assuming that overflow demand is Poisson distributed; specially

in the case that the fill rate is approximately 80%. Furthermore, we observe that the algorithm is more accurate when the number of warehouses is lower. This can be explained by the fact that in this scenario, there are less lateral transshipments, and in the extreme case that no lateral transshipments can occur, the algorithm is exact. The accuracy of the algorithm is not perfect, however, for practical purposes the performance is good enough. The computational times for the approximate evaluation algorithm are less than 0.001 seconds for each instance and therefore the algorithm is very suitable for using it in an optimization procedure.

We note that the accuracy of the approximate evaluation algorithm is less then the accuracy of the approximate evaluation procedure of Kranenburg and van Houtum [5]. Kranenburg and van Houtum study a special case of our model and for their main local warehouses they can evaluate the fraction of demand that is satisfied by an emergency shipment exactly, since complete pooling is applied. Our model is more general and therefore more difficult to evaluate.

	$\gamma(\mathbf{S}) \sim 0.8$		$\gamma(\mathbf{S}) \sim 0.9$		$\gamma(\mathbf{S}) \sim 0.95$	
scenario	$\overline{\Delta^e}$	$\Delta^e_{\max}$	$\overline{\Delta^e}$	$\Delta^e_{\max}$	$\overline{\Delta^e}$	$\Delta^e_{\max}$
6 warehouses	0.009	0.015	0.006	0.010	0.003	0.007
8 warehouses	0.013	0.019	0.010	0.020	0.007	0.014
10 warehouses	0.016	0.023	0.013	0.021	0.009	0.016
12 warehouses	0.020	0.030	0.016	0.028	0.011	0.022
14 warehouses	0.018	0.027	0.016	0.027	0.011	0.020

Table 3: Performance of the evaluation algorithm

5.4 Performance of the greedy algorithm

The performance of the greedy algorithm is compared with the optimal choice of base stock levels. Since determining the optimal base stock levels is computationally intensive, we have used two smaller test beds to evaluate the performance. The first test bed is based on Germany and the second test bed is based on France, where we isolated the demand for each country. In both countries we have selected 2, 3 and 4 local warehouses. We have omitted SKUs that have too high demand, for they require high base stock levels which is computationally expensive. For Germany we excluded SKUs 1,..,4 and 6,..,10, for France we excluded SKUs 1,2,3,5,6,8,9,16.

Since the evaluation algorithm always seems to overestimate $\gamma(\mathbf{S})$, it is possible that the greedy algorithm finds base stock levels that lead to a $\gamma(\mathbf{S}) < \gamma_0$. This makes it impossible to make a fair comparison with the optimal choice of base stock levels. We deal with this in the following way: we first set base stock levels with the greedy algorithm in which we used the approximate evaluation algorithm. Next we perform the exact evaluation procedure to check whether this choice of base

stock levels is feasible. If not, then we continue with the greedy algorithm, but then with the exact evaluation procedure to ensure a feasible solution.

Table 4 summarizes the results for Germany and Table 5 for France, where $\delta = (C(\mathbf{S})_{greedy} - C(\mathbf{S})_{exact})/C(\mathbf{S})_{exact}$ represent the relative cost difference, $\overline{\delta}$, σ_δ, δ_{max} correspond to the average, standard deviation and maximum of δ, respectively. The values are presented in percentages.

	2 warehouses			3 warehouses			4 warehouses		
γ_0	$\overline{\delta}$	σ_δ	δ_{max}	$\overline{\delta}$	σ_δ	δ_{max}	$\overline{\delta}$	σ_δ	δ_{max}
0.8	1.42	2.67	8.69	0.55	0.92	2.42	0.92	0.99	3.07
0.9	0.67	0.90	2.22	1.00	1.37	3.48	0.76	1.22	3.18
0.95	1.39	2.31	8.00	0.69	1.12	3.11	0.97	1.32	3.18

Table 4: Performance of the greedy algorithm: Germany

	2 warehouses			3 warehouses			4 warehouses		
γ_0	$\overline{\delta}$	σ_δ	δ_{max}	$\overline{\delta}$	σ_δ	δ_{max}	$\overline{\delta}$	σ_δ	δ_{max}
0.8	0.03	0.11	0.40	0.86	1.19	3.47	0.47	0.96	3.26
0.9	0.14	0.36	1.24	0.90	1.17	3.47	0.57	1.07	3.26
0.95	0.14	0.36	1.24	0.50	0.79	2.35	0.21	0.40	1.18

Table 5: Performance of the greedy algorithm: France

As we see from the tables, the costs related to the greedy algorithm are on average within 1.5% of the optimal costs for the case of Germany and 1% for France. The fact that the greedy algorithm performs worse in Germany can be explained by the fact that more lateral transshipments can take place, which make the approximate evaluation algorithm less accurate. Furthermore, we note that the performance of the algorithm does not deteriorate with the number of warehouses.

6. Managerial insights based on case study

In this section, we consider the situation of an OEM of high-tech equipment with the same data as used in Section 5. The aim of the experiments is to investigate the effect of taking lateral transshipments into account when setting base stock levels. We use 4 methods for setting base stock levels and compare them by simulation. In the simulation we simulate the network in steady state and compare the four methods under the same conditions. Regardless of the method used for determining the base stock levels, lateral transshipments take place in the execution of all simulation studies. The four methods are characterized as follows.

1) single-location planning without lateral transshipments;

2) single-location planning with lateral transshipments;

3) multi-location planning (with lateral transshipments);

4) multi-location planning with an adjusted time-based fill rate (and lateral transshipments).

Method 1 ignores lateral transshipments. In the optimization it is assumed that customers can be reached by at most one warehouse, so $\tilde{p}_n = \min\{p_n, 1\}\ \forall n \in N$. Under this assumption, the base stock level can be optimized separately for each warehouse. The approximate evaluation procedure given for the general case is exact in the special case that no lateral transshipments take place and the greedy method provides optimal base stock levels. Note that even though lateral transshipments are ignored while setting base stock levels, in the execution of the simulation study lateral transshipments can take place. Since Method 1 ignores the fact that lateral transshipments are possible in the network, the time-based fill rate $\gamma(\mathbf{S})$ is higher (in the soft sense) in the execution than it was planned while setting the base stock levels. Method 2 incorporates lateral transshipments by using an adjusted constraint on the time-based fill rate $\tilde{\gamma}_0$. We find $\tilde{\gamma}_0$ such that it is the lowest target that ensures a time-based fill rate $\gamma(\mathbf{S}) \geq \gamma_0$ in the execution. In order to find $\tilde{\gamma}_0$ we perform a bisection search using simulation to evaluate $\gamma(\mathbf{S})$, where we set base stock levels with the greedy algorithm using $\tilde{\gamma}_0$ for the time-based fill rate constraint.

Method 3 is the greedy algorithm as proposed in Section 4.2. Since we use the approximate evaluation method in the greedy algorithm, we cannot ensure that the time-based fill rate constraint is met. For a fair comparison we therefore introduce Method 4 that uses an adjusted time-based fill rate constraint. This adjusted time-based fill rate is the lowest adjusted constraint on the time-based fill rate $\tilde{\gamma}_0$ such that in execution $\gamma(\mathbf{S}) \geq \gamma_0$. We tune $\tilde{\gamma}_0$ by using simulation. We note that Method 2 and 4 use simulation and can require long computational times. Therefore, Methods 1 and 3 can be more attractive in practice.

We tested 11 scenarios for the number of warehouses, varying from 6 to 16. The scenarios are illustrated in Figure 3, where warehouses 1 through 6 are used in the first scenario, 1 through 7 in the second scenario, etc. For each scenario-SKU combination we evaluate each of the 4 methods by simulation. The time-based fill rate constraint is set equal to $\gamma_0 = 0.90$.

The results are summarized in Figures 4 to 7. Figure 4 shows the total cost over all SKUs and the overall time-based fill rates for all methods, where the overall time-based fill rate is defined as the demand weighted average of the fill rates over the SKUs. Figures 5, 6, and 7 show the costs and time-based fill rates for a cheap SKU (SKU 1), a medium-priced SKU (SKU 12), and an expensive SKU (SKU 20). From the overall data we conclude that Method 3 finds an infeasible solution in less than 7% of the cases, with a minimum time-based fill rate of $\gamma(\mathbf{S}) = 0.8855$. For the rest of this section we will only take Method 4 into account to ensure a fair comparison. Overall we see from the figures that Method 4 always outperform Methods 1 and 2, and Method 2 outperforms

Figure 3: Europe with 16 warehouses

Method 1.

SKU 1 is a cheap item and we observe in Figure 5 that all methods have the same performance. SKUs that are very cheap (like SKU 1) are expensive to deliver per emergency or lateral transshipment shipment compared to the inventory holding costs. Therefore, a high base stock level is desirable for this item at all warehouses and lateral transshipments will hardly ever occur. This is also obtained when we set base stock levels while lateral transshipments are not taken into account, and therefore the benefits of Method 4 are zero. SKU 12 is a medium-priced SKU and Figure 6 shows a large difference between Method 1 and Methods 2, 3, and 4. For a more expensive item, lateral transshipments are more important, since inventory holding costs are higher compared to the transportation costs. The time-based fill rate constraint in Method 1 forces the time-based fill rate to be much higher than required for the execution, which leads to high base stock levels and thus an expensive solution. When we cover this high time-based fill rate by tuning the constraint as in Method 2, we already see a big improvement. The base stock levels are not too high anymore, but we see that when lateral transshipments are explicitly taken into account as in Method 4, the improvements are even bigger. This is due to a better allocation of the stock over the warehouses.

We see the same effect for the very expensive SKU 20 in Figure 7. For SKU 20 the savings of Method 4 compared to the most simple Method 1 go up to 47%, and savings up to 19% can be gained when using Method 4 instead of Method 2. For SKU 20 the holding costs are dominant over the transportation costs, therefore it pays off to transport more, while sharing more inventory among the warehouses. The savings increase when the number of warehouses increase, because customers then have more options for delivery of an item, so a larger amount of inventory can be shared and the total inventory level can be decreased.

Furthermore we conclude from Figures 4 through 7 that the total costs decrease when the number of warehouses increase. For cheap items this is the case for both planning with and without lateral transshipments, but for expensive items, this is only the case when lateral transshipments are taken into account. For a network with both cheap and expensive items, the total costs decrease when the number of warehouses increases. When designing a network, fixed warehouse costs play a role as well. And when we plan with Method 1, the optimal number of warehouses is the minimal number required to cover the area such that the time-based service constraint can be met. However, when lateral transshipments are taken into account as in Method 4, then a higher number of warehouses might be optimal, depending on the costs of opening and operating warehouses. As in the case of the OEM we consider, fixed warehouse costs are very small compared to the inventory holding costs due to the fact that they work with logistic service providers, and for the design of their network, a larger number of warehouses is optimal. This is an important insight for the design of a spare parts network. For the design of spare parts network Candas and Kutanoglu [7] already showed that it is important to take inventory holding costs into account, we conclude that also lateral transshipments costs and emergency costs should be taken into account.

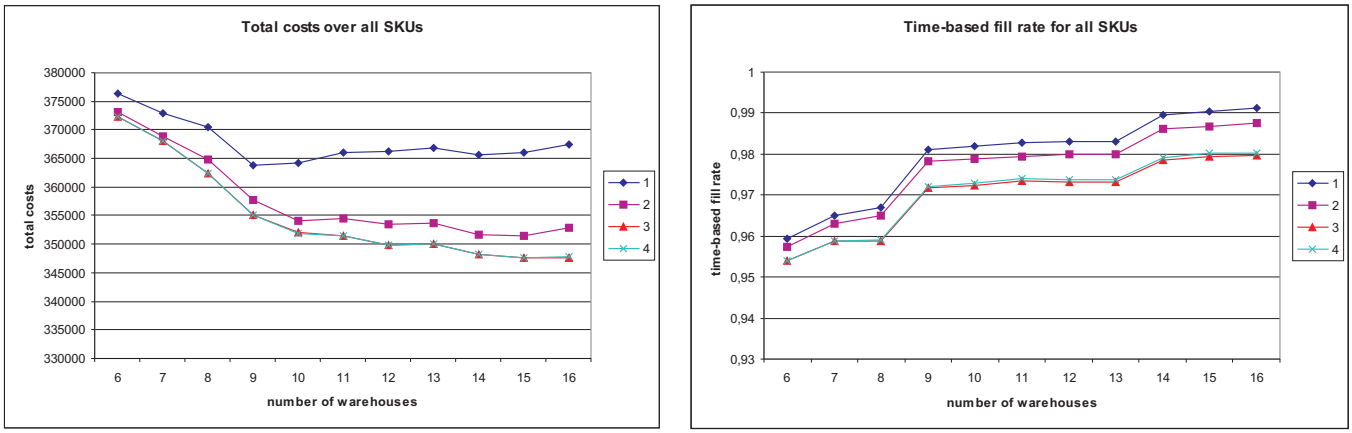
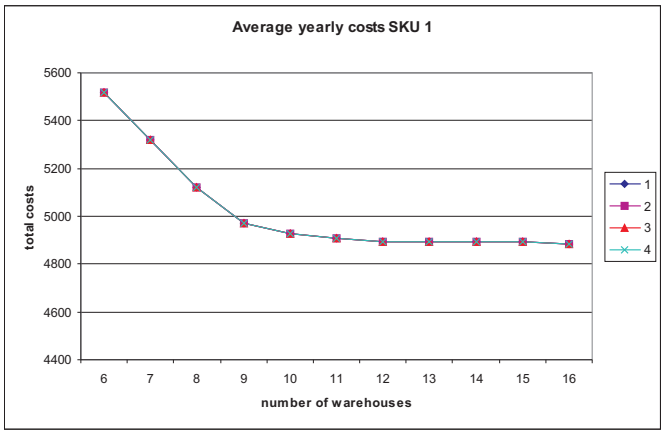


Figure 4: Performance over all SKUs



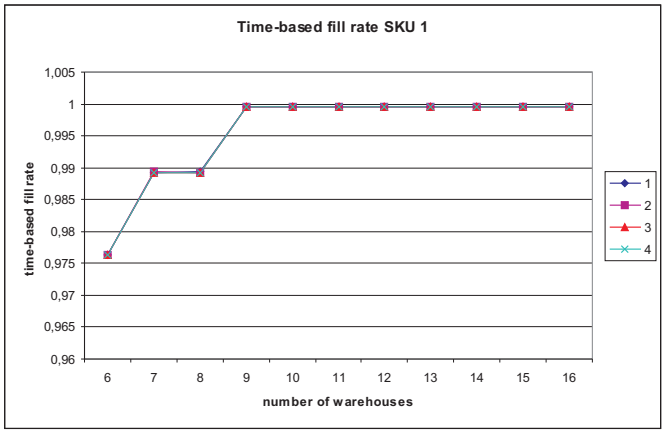


Figure 5: Performance for SKU 1

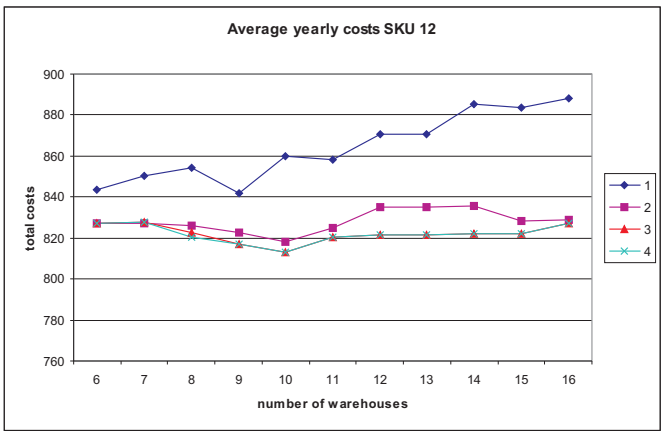


Figure 6: Performance for SKU 12

7. Extensions

The network model we proposed in Section 2 is a very flexible model that can handle time-based fill rate constraints. This model can be easily extended, adjusted to fit different networks (Section 7.1 and 7.2) or reformulated (Section 7.3) such that it fits different models as well. In this section we describe how the model can be extended, adjusted or reformulated such that it fits different models as well. Furthermore, we show how the approximate evaluation algorithm and the greedy algorithm work in these extensions.

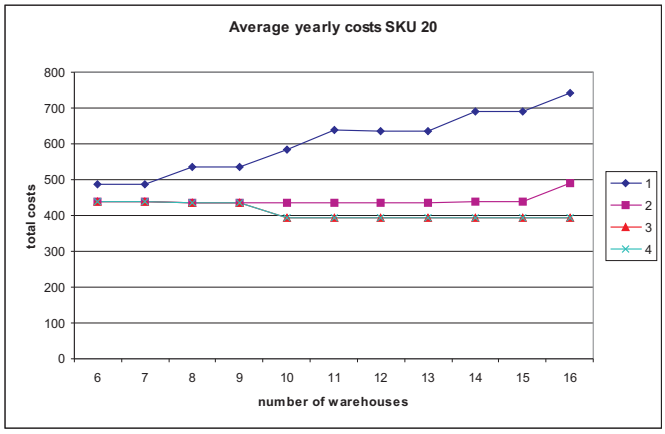


Figure 7: Performance for SKU 20

7.1 Random selection of warehouses for lateral transshipments

In the model presented we consider a fixed order for the delivery of lateral transshipments. However, in practice there can be situations that a random selection of warehouses for a lateral transshipment is more suitable. Random selection of warehouses for lateral transshipments has also been studied by e.g. Alfredsson and Verrijdt [1] and Axsäter [2]. The approximate evaluation algorithm can be easily adjusted for the situation that the warehouse to deliver the spare part via a lateral transshipment is randomly chosen amongst the set of reachable warehouses.

Assume that the first warehouse $v_n(1)$ for a customer is fixed, but there is no fixed order for lateral transshipments. So if a demand is generated at customer n then it will be delivered by warehouse $v_n(1)$, unless it is out of stock. In case it is out of stock, then an other warehouse is randomly chosen from the array, until there is a warehouse that can deliver, or all warehouses have been checked and none of them can deliver.

We describe by an example how the approximate evaluation algorithm can be adjusted to deal with the random selection of warehouses for the use of lateral transshipments. Consider a customer n with 3 warehouses that can reach the customer within the time limit, where $v_n(1) = 1$, $v_n(2) = 2$, and $v_n(3) = 3$. The demand that warehouse 1 faces from customer n is Poisson distributed with mean μ_n, just as before. Demand that warehouse 2 faces from customer n is now a bit more complex, since there are now two different ways that demand can end up at warehouse 2. When warehouse 1 is out of stock, there is a probability 1/2 that warehouse 2 is asked for a lateral transshipment directly. The same holds for warehouse 3. When warehouse 3 is asked for a lateral transshipment and it is out of stock, the demand is redirected to warehouse 2. So the average total demand warehouse 2 faces from customer n equals:

$$M_{n,2} = \frac{1}{2}(1-\beta_1)\mu_n + (1-\beta_1)(1-\beta_3)\mu_n$$

Similar formulas can be derived for warehouse 3. By using the corrected formulas for $M_{n,\upsilon_n(i)}$, the same approximate evaluation algorithm can be used to determine $\alpha_{n,j}(\mathbf{S})$ and $\theta_n(\mathbf{S})$. This approximation can be implemented in the greedy heuristic.

We expect similar results for the performance of the adjusted approximate evaluation algorithm and the greedy heuristic, since the approximate evaluation algorithm is based on the same assumptions, namely that the overflow demand is Poisson distributed and the base stock levels of the warehouses are independent of each other.

7.2 Multi-item model

For the case of the OEM we consider, we are faced with a service requirement that is set per SKU, however, in practice we often encounter a service requirement over all SKUs. In that case a multi-item model would be more suitable. For the evaluation algorithm this does not make a difference, one can evaluate the performance of the different items separately, but for the determination of the base stock levels, small adjustments have to be made. First of all, an aggregated time-based fill rate has to be defined that measures the fraction of all demand that is satisfied within the time limit. Next, in the greedy heuristic, one has to replace the time-based fill rate by the aggregated time-based fill rate, and the costs per item by the costs for all items. Finally, in both loops of the greedy algorithm, the SKU-warehouse combination should be find that minimizes the (relative) additional costs.

7.3 Car stock

The network model as we described is motivated by the network of an OEM of high-tech equipment. At another OEM we encountered a different network that can be reformulated such that our methods are applicable as well. At this OEM spare parts are kept on stock in several warehouses spread over the geographical area where the customers are situated. In addition, service engineers, who drive around in cars to visit the customers for both preventive and corrective maintenance, carry some so-called "car stock" in their cars. In case a customer requires a spare part and the engineer is out of stock for that part, then an emergency transshipment is requested from a warehouse close by. Depending on the location of the customer, the engineer will request a spare part at nearby warehouses in a fixed order. So, each customer has a fixed order in which it asks for a lateral transshipment as in the model we described. Note that the delivery time of the spare part is measured from the moment that the service engineer diagnosed the a certain part has failed, so

the travel time of the service engineer is not included in the delivery time. For this network we can apply our methods as well.

8. Conclusion

In this paper, we consider a spare parts inventory network consisting of multiple local warehouses and a central warehouse under delivery time requirements. In this network lateral transshipments are allowed, but only if the lateral transshipment can reach the customer within a pre-specified time limit. The problem is to set base stock levels for the spare parts at the local warehouses and gain insight in the savings that can be gained when lateral transshipments are taken into account in the planning phase. We develop an approximate evaluation algorithm that can evaluate the network under given base stock levels. This approximate evaluation method is used in a greedy algorithm to set base stock levels at the local warehouses.

The results indicate that taking lateral transshipments into account can lead to significant cost reductions for expensive spare parts. Furthermore, the results indicate that higher savings can be gained with a higher number of warehouses. Costs decrease when the number of warehouses increases, while this is not the case when lateral transshipments are not taken into account in the planning. The model we introduced is applicable in many real-life cases and the methods we proposed can be easily extended or adjusted.

References

[1] Alfredsson, P., Verrijdt, J. 1999. *Modeling Emergency Supply Flexibility in a Two-Echelon Inventory System*. Management Science Vol. 45 No. 10, pp. 1416 – 1431.

[2] Axsäter, S. 1990. *Modelling emergency lateral transshipments in inventory systems*. Management Science Vol. 36 No. 11, pp. 1329–1338.

[3] Axsäter, S. 2003. *Evaluation of unidirectional lateral transshipments and substitutions in inventory systems*. European Journal of Operational Research Vol. 149 No. 2, pp. 438–447.

[4] Caggiano, K.E., Jackson, P.L., Muckstadt, J.A., Rappold, J.A., 2009. *Efficient computation of time-based customer service levels in a multi-item, multi-echelon supply chain: A practical approach for inventory optimization*. European Journal of Operational Research, Vol. 199, No. 3, pp. 744–749

[5] Kranenburg, A.A., van Houtum, G.J. 2009. A new partial pooling structure for spare parts networks. European Journal of Operational Research, Vol 199, No. 3, pp. 908–921 .

[6] Kukreja, A., Schmidt, C.P., Miller, D.M. 2001. *Stocking Decisions for Low-Usage Items in a Multilocation Inventory System*. Management Science, Vol. 47 No. 10, pp. 1371-1383.

[7] Candas, M.F., Kutanoglu, A. 2007. *Benefits of considering inventory in service parts logistics network design problems with time-based service constraints* IIE Transactions, Vol. 39, pp. 159-176.

[8] Kutanoglu, E. 2008. *Insights into inventory sharing in service parts logistics systems with time-based service levels.* Computers & Industrial Engineering 54, pp. 341–358.

[9] Kutanoglu, E., Mahajan, M. *An inventory sharing and allocation method for a multi-location service parts logistics network with time-based service levels.* European Journal of Operational Research, Vol. 194, pp. 728–742.

[10] Lee, H.L. 1987. *A multi-echelon inventory model for repairable items with emergency lateral transshipments.* Management Science Vol. 33 No. 10, pp. 1302–1316.

[11] Liu, J., Lee, C.G. 2007 *Evaluation of inventory policies with unidirectional substitutions.* European Journal of Operational Research Vol. 182, No. 1, pp. 145–163

[12] Olsson, F. 2009 *An inventory model with unidirectional lateral transshipments.* European Journal of Operational Research, doi:10.1016/j.ejor.2009.01.024.

[13] Tagaras, G., Cohen, M.A. 1992. *Pooling in Two-Location Inventory Systems with non-negligible replenishment leadtimes.* Management Science Vol. 38 No. 8, pp. 1067 – 1083.

[14] Tijms, H.C., 2003. *A First Course in Stochastic Models.* Wiley, Chichester, England.

[15] Wong, H., van Houtum, G.J., Cattrysse, D., van Oudheusden, D. 2005. *Simple efficient heuristics in multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints.* The Journal of the Operational Research Society Vol. 56 No. 12. pp. 1419 – 1431.

[16] Wong, H., Cattrysse, D. , van Oudheusden, D. 2005. Inventory pooling of repairable spare parts with non-zero lateral transshipment time and delayed lateral transshipments . European Journal of Operational Research 165, 207 – 218.

[17] Wong, H., van Houtum, G.J., Cattrysse, D. van Oudheusden, D. 2006. *Multi-item spare parts systems with lateral transshipments and waiting time constraints.* European Journal of Operational Research 171. pp. 1071 – 1093.

[18] Zhao, H., Deshpande, V., Ryan, J.K. 2006. *Inventory Sharing and Rationing in Decentralized Dealer Networks.* Management Science 51, pp. 531 – 547.