A system approach for multi-location spare parts systems with lateral transshipments and waiting time constraints

A System Approach for Multi-Location Spare Parts Systems with Lateral Transshipments and Waiting Time Constraints

H. Wong [a], G. J. van Houtum [b], D. Cattrysse [a] and D. Van Oudheusden [a]

[a] *Katholieke Universiteit Leuven, Centre for Industrial Management, Celestijnenlaan 300A, 3001 Heverlee, Belgium*
[b] *Technische Universiteit Eindhoven , Faculty of Technology Management, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

Abstract

This paper deals with the analysis of a multi-item, continuous review model of a multi-location inventory system for repairable spare parts, in which lateral and emergency shipments occur in response to stock-outs. A continuous review base-stock policy is assumed for the inventory control of the spare parts. The objective is to minimize the total cost for inventory holding, lateral transshipments and emergency shipments subject to a target level for the average waiting times at all locations. We structure the optimization problem as a combinatorial problem and four different heuristics are developed and evaluated in terms of their total costs and computation times. A lower bound obtained by applying Lagrangian relaxation is used for the evaluation of the heuristics. A first computational experiment shows that the greedy-type heuristic has the best performance, with an average gap to the lower bound of only 0.61%. Interesting insights with respect to the relative cost savings achieved by applying a system approach and pooling policy are obtained from a second experiment.

Keywords: inventory; spare parts; system approach; lateral transshipment; heuristic

Introduction

Equipment-intensive industries such as airlines, nuclear power plants, various process and manufacturing plants using complex machines often require large quantities of spare parts to guarantee high system availability which in turn results in excessive holding cost. The aviation industry for example, must carry about $32 billion each year to stock the spare parts they need to keep their airplanes flying.[1] Unfortunately, there can be massive inefficiencies in how such inventories are utilized. On the one hand, companies can find themselves carrying an excessive number of spare parts. On the other hand, if they were not available when needed, companies will face severe downtime consequences. As many parts are very expensive, critically important and their failure rates are so low that they are difficult to forecast, spare parts inventory management within these industries is one of the hardest problems to deal with. Traditionally, companies relied on the recommended spare parts listings provided by the Original Equipment Manufacturer (OEM). Today, companies need to plan more smartly to cut down their inventory cost.

It is well known from the literature on spare parts inventory management that the system approach and the lateral transshipment (inventory pooling) policy are two approaches that may lead to significant reductions in inventory cost. Under a system (multi-item) approach, all parts in the system are considered when making inventory-level decisions. In contrast, under an item approach, inventory levels for each individual part are set independently. Previous studies, e.g. Sherbrooke,[2] Thoneman *et al*,[3] and Rustenburg *et al*[4] have shown that a system approach gives significant savings in inventory cost in comparison to an item approach. Cooperation among companies through inventory pooling or lateral transshipments can also be used to improve the companies' service levels while reducing the total system cost at the same time. Lateral transshipments are used to satisfy a demand at a location that is out of stock from another location with a surplus of on-hand inventory. In case of a stock-out, a lateral transshipment is certainly preferable to an emergency shipment as long as the cost and lead-time of a lateral transshipment are lower than the ones of an emergency shipment. In a system where the downtime has to be as low as possible, lateral transshipments are therefore very important in reducing the system's downtime and inventory cost.

Most of the work related to spare parts systems with lateral transshipments has been done in the context of single-item problems. Lee[5] considers lateral transshipments in a two-echelon inventory system for repairable items that employs continuous-review inventory policy. He analyzes a system which consists of one depot and several bases. The bases are grouped into several pooling groups such that members of each group are identical with respect to the stock level and demand rate. He developed a model to derive an approximation for the fractions of demand satisfied immediately from stock, demand satisfied by lateral transshipment, and demand that is backordered. Axsäter[6] improves Lee's model by relaxing the assumption that bases have to be identical. His model puts more emphasis on modeling the demand processes at the bases. He models the effective demand rate at a warehouse under two conditions: when stock on hand is positive and when stock on hand is zero or negative. When stock on hand is positive, a base faces its own demand and lateral transshipment requests from other bases.

When stock on hand is zero or negative, the only demand considered is the demand that has to be backordered. A random sourcing rule is used to select a base as the source of lateral transshipment. Sherbrooke[7] presents a simulation study to investigate the importance of lateral transshipments in a two-echelon depot-base system for repairable items. Regression analysis on the simulation data is used to derive approximate expressions for the expected system backorders. Several authors develop new models based on the model of Axsäter.[6] Alfredsson and Verrijdt[8] extend Axsäter's model by allowing emergency shipments from the depot and an outside supplier. In the case when all bases are out of stock, the depot has stock on hand, and a demand arrives at any base, the demand is satisfied through direct delivery from the depot. If there is no stock on hand at the depot, the demand is satisfied by direct delivery from an outside supplier. With such emergency shipments, no demands arriving at any bases are backordered. In this paper we also consider such emergency shipments that are commonly practiced in many real-life situations, especially when the downtimes have to be as low as possible. Like Axsäter,[6] Alfredsson and Verrijdt[8] assume exponential lead times, and, based on their simulation results, they conclude that the lead time distribution hardly affects the service performance. In a more recent work, Kukreja *et al*[9] generalize Axsäter's model by relaxing the assumption of an exponential replenishment (repair) lead time distribution. They also prove that the service performance is almost identical for all lead time distributions with the same finite mean. This finding is important since our approach will be based on Markov analysis and the assumption of exponential lead-time distribution is therefore required. Other papers consider lateral transshipments under periodic review policies. Examples are Gross,[10] Krishnan and Rao,[11] Das,[12] Karmarkar,[13] Tagaras,[14,15] Robinson,[16] Tagaras and Cohen,[17] and Herer and Rashit;[18] see Wong *et al*,[19] for an overview of these papers.

The work by Archibald *et al*[20] and our previous work[19] are the only existing studies addressing lateral transshipments in the context of multi-item problems. Archibald *et al*[20] analyze a multi-period, periodic-review model of a *two-location* inventory system in which lateral transshipments can occur at any time during the period. They first formulate the two-location, single-item inventory problem as a Markov decision process and they then extend the results to a two-location, multi-item inventory problem with limited storage space. Different from their work, Wong *et al*[19] analyze a multi-item, continuous-review model of a *two-location* inventory system for repairable spare parts in which lateral and emergency shipments can occur in response to stock-outs and there is a target level for the average waiting time (for all items together) at each of the two locations. A solution procedure was developed based on Lagrangian relaxation that gives both a heuristic solution and a lower bound for the optimal total cost. The performance of this solution procedure is quite satisfactory as very small gaps (below one percent) were obtained between the cost under the heuristic solution and the lower bound. This solution procedure, however, has a limitation since it requires a long computation time to solve rather large problems. For example, an average computation time of 14 minutes is required to solve two-location problems with 100 items (run on a PC with a 333-MHz Pentium II processor). As the computation time can increase very fast when more locations are involved, it would be useful to develop a more efficient solution procedure.

In this paper, we address multi-item problems with lateral transshipments for the case with *multiple locations*. We develop a more efficient, but still accurate solution procedure for solving the problems that involve many items and several locations. Such solution procedure enables us to analyze savings that can be obtained through the application of pooling and system approach in systems with up to hundred items and six locations.

The paper is organized as follows. In Section 2, we present the problem formulation. We introduce the basic assumptions and the notations of the model, and we present the mathematical formulation of our problem. In Section 3 we describe a local search based solution procedure and develop three different heuristics. The Lagrangian relaxation approach for obtaining the lower bound of the optimal total cost is also described. The lower bounds are needed for the evaluation of the heuristics. Section 4 presents our computational experiment for the evaluation of the heuristics. Further, we also analyze the cost savings that are obtained by applying a system approach and pooling policy in comparison to an item approach and no-pooling policy. Finally, we summarize the results in Section 5 and conclude with directions for further research.

Model description

In this section we introduce our model. Firstly, we present the assumptions and notations used in the model. Secondly, we describe the model for the evaluation of a stocking policy and lastly, we formulate the optimization problem.

Assumptions and notations

We model the situation of J ($J \geq 2$) independent companies who keep spare parts on stock for their technical systems. Note that throughout this paper the words company and location are used interchangeably. The companies are indexed by j = 1, 2, …, J. We assume that all companies have a number of technical systems of the same type. These systems consist of components which are subject to failures. In total there are I different items (SKUs). These items are indexed by i = 1, 2, …, I. Failures occur according to Poisson processes with constant rates. The total failure rate of components of item i at company j is given by m_j^i ($\geq$ 0). If an item i does not occur in the configurations of the technical systems at company j, then $m_j^i = 0$. We assume that $\sum_{j=1}^{J} m_j^i > 0$ for all i. Further, $M_j = \sum_{i=1}^{I} m_j^i$ denotes the total failure rate at company j. We assume that $M_j > 0$ for $j = 1,\ldots, J$. All companies use base-stock policies for the inventory control. Company j has $S_j^i (\in \mathbb{N}_0 := \{0\} \cup \mathbb{N})$ spare parts of item i. In total, the companies share S_{tot}^i spare parts of item i, with $S_{tot}^i = \sum_{j=1}^{J} S_j^i$.

All parts are repairable and there is no condemnation. When a part of item i fails at company j, the failed part is immediately removed and sent into repair. A ready-for-use part is put back into the system where the failure occurred as soon as such a part is available. If company j has a ready-for-use part on stock then this can be done immediately. If not, then there is a waiting time for a ready-for-use part. In that case, the required part may be obtained by a lateral transshipment from another company that has a ready-for-use part on stock. The waiting time is then limited to the average transshipment time TL_{jk}^i (= TL_{kj}^i) where k is the company selected as the source of the lateral transshipment. Since it is possible to have two or more companies for the source of a lateral transshipment, a selection rule is required. In this paper we use the closest neighbor sourcing rule that is also used by Kukreja *et al*,[9] and ties are broken with equal probabilities. Wong[21] shows that the closest neighbor sourcing rule is preferable to the random sourcing rule used by Axsäter[6] and Alfredsson and Verrijdt.[8] Further, we also assume that complete pooling is applied. That means a company offers its entire available inventory when there is a lateral transshipment request from another company experiencing a stock-out. If no ready-for-use part is available at any of the companies, then an emergency supply mode is applied. This means that either the repair operation is expedited or the required part is ordered from an outside supplier e.g., an OEM or a third party supplier. A ready-for-use part then becomes available after an average time TE^i . Failed parts that are sent into repair are returned as ready-for-use parts after exponential repair lead-times. The lead-times of different parts of the same item and of parts of different items are independent. The repair rate of a failed part of item i is given by μ^i. We assume that in case a lateral transshipment (an emergency shipment) takes place from company j (the outside supplier), the failed part will be returned to company j (the outside supplier) upon completion of its repair. With this assumption, the number of parts on stock plus the number of parts in repair of item i at company j is always equal to S_j^i.

At company j, there is a maximum level W_j^{max} given for the average waiting time per request for a ready-for-use part. This average is calculated for all items together. Total system cost consists of holding cost, transshipment cost and emergency shipment cost. A holding cost CH^i is counted for each spare part of item i. The transshipment cost is dependent on the locations between which the lateral transshipment takes place. For item i, a cost CT^i is counted for each distance unit of lateral transshipment. A cost CE^i is counted for each part coming from the emergency supply. The objective is to find a stocking policy $\underline{S} = (\underline{S}^1;...;\underline{S}^I) = (S_1^1, S_2^1, ..., S_J^1;\ S_1^2, S_2^2, ..., S_J^2;\ ...\ ;\ S_1^I, S_2^I, ..., S_J^I)$ under which the average total cost is minimized subject to the waiting time constraints for all companies.

Exact evaluation of a given policy

We now describe an exact evaluation of a stocking policy. We first need to define:

β_j^i = fraction of demands for item i at company j satisfied by company j itself

α_{jk}^i = fraction of demands for item i at company j satisfied by lateral transshipments from company k ($k \neq j$)

α_j^i = $\sum_{k=1,k\neq j}^{J} \alpha_{jk}^i$ fraction of demands for item i at company j satisfied by lateral transshipments

θ_j^i = fraction of demands for item i at company j satisfied by emergency supplies

W_j = average waiting time per request for a ready-for-use part at company j

Obviously, $\beta_j^i + \alpha_j^i + \theta_j^i = 1$ for $i = 1,\ldots,I$; $j = 1,\ldots J$. Since complete pooling is applied here, θ_j^i is the same for $j = 1,\ldots J$, i.e. $\theta_1^i = \theta_2^i = ... = \theta_J^i = \theta^i$ for all i.

The system behavior with respect to an item i is independent of all other items and can be described by a J-dimensional Markov process. For each item i, we introduce the state $\boldsymbol{x}^i = (x_1^i, ..., x_J^i)$, where x_j^i represents the physical stocks of spare parts of item i at company j, and $0 \leq x_j^i \leq S_j^i$, $x_j^i \in \mathbb{N}_0$. We define $\boldsymbol{x}_{j+}^i = (x_1^i, ..., x_{j-1}^i, x_j^i + 1, x_{j+1}^i, ..., x_J^i)$ and $\boldsymbol{x}_{j-}^i = (x_1^i, ..., x_{j-1}^i, x_j^i - 1, x_{j+1}^i, ..., x_J^i)$. All possible transitions of the Markov process are as follows:

Transition 1: a failure of a part of item i occurs at company j while $x_j^i > 0$; the state transition is $\boldsymbol{x}^i \rightarrow \boldsymbol{x}_{j-}^i$; The transition rate is m_j^i.

Transition 2: a failure of a part of item i occurs at company j while $x_j^i = 0$ and $x_k^i > 0$ for at least one other company $k \neq j$. Define $K = \left\{k \mid k \neq j, x_k^i > 0, T_{jk}^i \leq T_{jm}^i \text{ for all } m \neq j\right\}$. A company $k \in K$ is selected as the source of the lateral transshipment with probability $1/|K|$. The state transition is $\boldsymbol{x}_k^i \rightarrow \boldsymbol{x}_{k-}^i$. The transition rate is $m_j^i/|K|$ and this represents a lateral transshipment sent from company k to company j.

Transition 3: a failure of a part of item i occurs at company j while $\boldsymbol{x}^i = \boldsymbol{0}$. An emergency supply is applied; the state remains $\boldsymbol{0}$. The transition rate is m_j^i.

Transition 4: the repair of a part of item i belonging to company j is completed. The state transition is $\boldsymbol{x}^i \rightarrow \boldsymbol{x}_{j+}^i$. The transition rate is $(S_j^i - x_j^i)\mu^i$.

Figure 1 shows an example of the Markov process for a system with J = 3, (S_1^i, S_2^i, S_3^i) = (2,1,1), $TL_{23}^i < TL_{12}^i < TL_{13}^i$.

Let $q_{x \rightarrow x'}$ denote the transition rate from state $\boldsymbol{x}$ to state $\boldsymbol{x'}$. All the defined transition rates will form an infinitesimal generator Q of an irreducible continuous-time Markov chain and Q has the size of N x N, where $N = \prod_{j=1}^{J}(S_j^i + 1)$ represents the total number of states in the Markov process. We define $\boldsymbol{\pi}$ as the steady-state probability vector and $\boldsymbol{\pi}$ is determined by solving $\boldsymbol{\pi}Q = 0$.

The fraction of demands for item i satisfied by emergency supplies is equal to the probability of being in state $\boldsymbol{0}$. Thus, we can write $\theta^i = \pi_{\boldsymbol{0}}$. This fraction can also be obtained by the aggregation on the basis of total physical stock at all locations. This shows that θ^i is also equal to the Erlang loss probability of an $M/M/S_{tot}^i/S_{tot}^i$ queuing system.

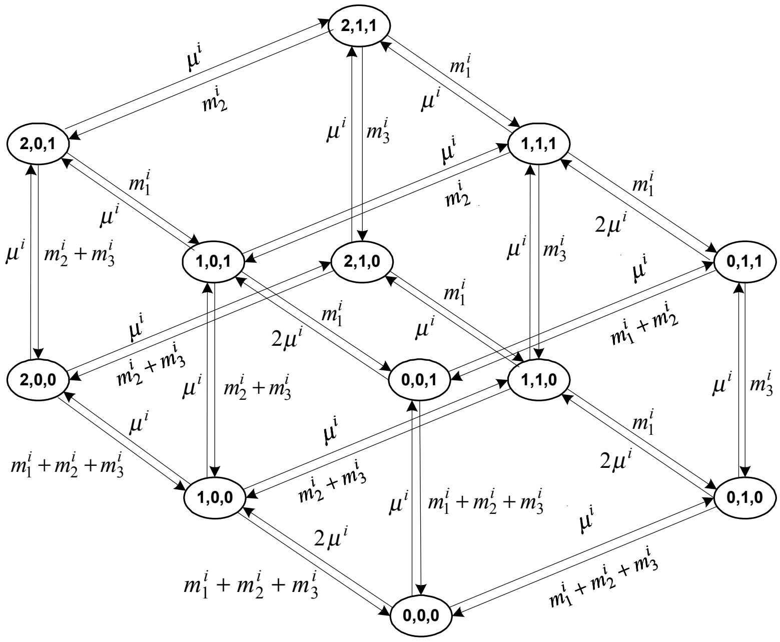

Figure 1. Markov process for (S_1^i, S_2^i, S_3^i) = (2,1,1)

Next, we show how to determine the fraction of demands satisfied by lateral transshipments. Suppose we want to determine α_{jk}^i, the fraction of demands for part of item i at company j satisfied by lateral transshipments from company k. We first define

$$X_{jk}^i = \left\{ x^i \mid x_j^i = 0, x_k^i > 0, x_m = 0 \text{ for all } m \neq j, m \neq k, TL_{jm}^i < TL_{jk}^i \right\},$$

$$Y_{jk}^i(x^i) = \left\{ m \mid m \neq j, m \neq k, x_m^i > 0, TL_{jm}^i = TL_{jk}^i \right\} \forall x^i \in X_{jk}^i .$$

We can then determine α_{jk}^i using the following expression:

$$\alpha_{jk}^i = \sum_{x^i \in X_{jk}^i} 1/(1+|Y_{jk}^i(x^i)|)\pi_{x^i} . \tag{1}$$

The expected waiting time for a lateral transshipment of a part of item i to company j is given by $D_j^i = \sum_{k=1,k\neq j}^{J} \alpha_{jk}^i TL_{jk}^i$. The fraction of demands satisfied by local stock equals to $\beta_j^i = 1 - \alpha_j^i - \theta^i$.

We now explain how to calculate the average waiting time. When there is a failure of a part of item *i*, the waiting time for a ready-for-use part at company *j* is given by:

$$W_j^i(\underline{S}^i) = \beta_j^i 0 + \sum_{k=1,k\neq j}^{J} \alpha_{jk}^i TL_{jk}^i + \theta^i TE^i$$

$$= D_j^i + \theta^i TE^i \qquad (2)$$

Taking all items together, the average waiting time per request for a ready-for-use part at company *j* for a given stocking policy can be expressed as:

$W_j(\underline{S}) = \sum_{i=1}^{I}$ *Prob* {an arbitrary failing part at location *j* is of item *i*} (average waiting time for a ready-for-use part of item *i*)

$$= \sum_{i=1}^{I} \frac{m_j^i}{M_j} W_j^i(\underline{S}^i)$$

$$= \sum_{i=1}^{I} \frac{m_j^i}{M_j} (D_j^i + \theta^i TE^i) \qquad (3)$$

The optimization formulation

With all the above defined notations, we can now formulate our optimization problem as follows:

Problem $\boldsymbol{P_0}$: Minimize $Z(\underline{S}) = \sum_{j=1}^{J} \sum_{i=1}^{I} \left(CH^i S_j^i + CT^i m_j^i D_j^i + CE^i m_j^i \theta^i \right) \qquad (4)$

subject to $\sum_{i=1}^{I} \frac{m_j^i}{M_j} (D_j^i + \theta^i TE^i) \leq W_j^{max} \quad j = 1, \ldots, J \qquad (5)$

$S_{ij} \in \mathbb{N}_0 \quad i = 1, \ldots, I; \ j = 1, \ldots, J. \qquad (6)$

Problem $\boldsymbol{P_0}$ is an integer-programming problem with a non-linear objective function and non-linear constraints.

Description of the heuristics

We can structure problem $\boldsymbol{P_0}$ as a combinatorial problem. The goal is to find a stocking policy $\underline{S} \in \mathbb{S}$ that minimizes the total cost where $\mathbb{S}$ is the set of all *feasible* solutions for problem $\boldsymbol{P_0}$. We develop a solution procedure that is based on local search, also referred to as neighborhood search. A basic local search algorithm begins with an arbitrary feasible solution and it tries to improve this solution by making small changes to it and ends up in a local minimum where no further improvement is possible. Following the general local search optimization, our solution procedure consists of two steps. The first step generates an initial solution and the second step executes the local search process. As the choice of an initial solution may influence the quality of the final outcome, we will evaluate three different initialization algorithms and apply one and the same local search process for all initialization algorithms.

Generation of an initial solution

Initialization algorithm 1 (greedy approach)

This procedure is iterative. We start with $\underline{S} = \mathbf{0}$ and next, in each iteration one unit of stock is added into the system. The iteration process is terminated when a feasible solution is obtained. In each iteration, we add one unit of stock for the item and the location that gives the largest decrease in distance to the set of feasible solutions per extra unit of costs. Let $e_j^i = (0,...,0,1,0,...,0)$ denote a stocking policy with one unit of stock for item i and location j and zero stocks for all other items and locations. We define for each solution $\underline{S}$ the distance to the set of feasible solutions as $\sum_{j=1}^{J}(W_j(\underline{S}) - W_j^{\max})^+$ where $(a)^+ = \max(0,a)$. For each combination of $i \in \{1,...,I\}$ and $j \in \{1,...,J\}$, we calculate the ratio $r_j^i = \dfrac{\Delta W_j^i}{\Delta Z_j^i}$ where:

$$\Delta W_j^i = \sum_{j=1}^{J}(W_j(\underline{S}) - W_j^{\max})^+ - \sum_{j=1}^{J}(W_j(\underline{S} + e_j^i) - W_j^{\max})^+ \qquad (7)$$

and

$$\Delta Z_j^i = Z(\underline{S} + e_j^i) - Z(\underline{S}) \qquad (8)$$

One unit of stock is then added for the item and location with the highest ratio. The formal statement of this initialization procedure is as follows:

Initialization algorithm 1

Step 1: Set the initial solution $\underline{S} = \mathbf{0}$; calculate $W_j(\mathbf{0})$ for all locations *j*.

Step 2: For all $i \in \{1,\ldots, I\}$ and $j \in \{1,\ldots, J\}$: Calculate ΔW_j^i, ΔZ_j^i, and r_j^i.

Step 3: Let *i** and *j** be the combination with the highest ratio. Set $\underline{S} = \underline{S} + e_{j*}^{i*}$.

If $W_j(\underline{S}) \le W_j^{\max}$ for all *j* go to *END*; otherwise go to *Step* 2 .

END

Notice that in the above algorithm, during the first iteration we need to solve in total IJ Markov processes to obtain the values of ΔW_j^i and ΔZ_j^i for all combinations of *i* and *j*. All values are kept in the memory so that for the next iterations we only need to solve the new Markov process for the item of which the number of stock has been changed. (Recall that any changes of the stock levels for an item do not affect the Markov processes of all other items). This is important in minimizing the computation time of the algorithm since solving the Markov processes constitutes the most time consuming part of the algorithm.

Initialization algorithm 2 (item approach)

In this initialization procedure the stocking policy is determined for each item independently. For each item we have the problem of minimizing the total cost while the average waiting time per request for a ready-for-use part of this item can meet the target maximum waiting times at all locations. Intuitively, the solution for each individual item will form a feasible solution for the original problem. To solve the problem for each item *i*, we use a similar method as in the first initialization procedure. We start with zero stock at all locations. Then we increase the stock incrementally by one until a feasible solution is obtained. For each iteration, one unit of stock is added at the location that gives the largest decrease in distance to the set of feasible solutions per extra unit of costs. The formal statement of this procedure is as follows:

Initialization algorithm 2

(This algorithm is applied for each item *i*)

Step 1: Set the initial solution $S^i = \mathbf{0}$; calculate $W_j^i(\underline{S}^i)$ for all locations *j*.

Step 2: For all $j \in \{1,\ldots, J\}$: Calculate ΔW_j^i, ΔZ_j^i, and r_j^i.

Step 3: Let *j** be the location with the highest ratio. Set $S_{j*}^i = S_{j*}^i + 1$.

If $W_j^i(\underline{S}^i) \le W_j^{\max}$ for all *j* go to *END*; otherwise go to *Step* 2 .

END

Initialization Procedure 3 (no-pooling approach)

In this initialization procedure, lateral transshipments are not allowed among locations. This implies that the optimization problem for each location is solved independently. We apply a Lagrangian relaxation approach to solve this problem. For a given Lagrange multiplier λ , the relaxed problem for each company *j* can be formulated as follows:

$$\text{Minimize } \sum_{i=1}^{I}\left(CH^i S_{tot}^i + CE^i m_j^i \theta^i(S_{tot}^i)\right) + \lambda\left(\sum_{i=1}^{I} m_j^i \theta^i(S_{tot}^i) TE^i - M_j W_j^{\max}\right)$$

The optimal Lagrange multiplier for this problem can be easily solved using a standard bisection procedure. A more detail description of this procedure is described in Wong *et al*.[19] Since we have a closed-form formula for the Erlang loss probability, no numerical solutions of Markov chains are needed here. As a result, we could expect that this initialization procedure is faster than the other two procedures. But with respect to the quality of the obtained initial solution, this initialization procedure would probably give the most expensive initial solution since higher stock levels are required with the absence of pooling.

Local search

We will first define the neighborhood structure for our problem. For each solution $\underline{S}$, we define the neighborhood of $\underline{S}$ as $N(\underline{S}) = N_1(\underline{S}) \cup N_2(\underline{S}) \cup N_3(\underline{S}) \cup N_4(\underline{S})$ where:

$$N_1(\underline{S}) = \left\{\text{all } \underline{S}' \in \mathbb{S} \mid \underline{S}' = \underline{S} - e_j^i, i \in \{1,...,I\}, j \in \{1,...,J\}\right\}$$

$$N_2(\underline{S}) = \left\{\text{all } \underline{S}' \in \mathbb{S} \mid \underline{S}' = \underline{S} + e_j^i, i \in \{1,...,I\}, j \in \{1,...,J\}\right\}$$

$$N_3(\underline{S}) = \left\{\text{all } \underline{S}' \in \mathbb{S} \mid \underline{S}' = \underline{S} + e_j^i - e_{j'}^{i'},\ i \in \{1,...,I\}, i' \in \{1,...,I\}, j \in \{1,...,J\}, j' \in \{1,...,J\}, i \ne i'\right\}$$

$$N_4(\underline{S}) = \left\{\text{all } \underline{S}' \in \mathbb{S} \mid \underline{S}' = \underline{S} + e_j^i - e_{j'}^i, i \in \{1,...,I\}, j \in \{1,...,J\}, j' \in \{1,...,J\}, j \ne j'\right\}$$

The neighborhood of a solution can thus be seen as an integration of four sub-neighborhoods. The first sub-neighborhood is formed by reducing one unit of stock for each combination of *i* and j. In most cases when the transportation cost is small compared to the inventory holding cost, moving to this neighborhood may be useful in order to reduce the inventory holding cost. In contrast, moving to a neighbor of the second sub-neighborhood which is formed by adding one unit of stock, may be useful when the transportation cost is very expensive compared to the inventory holding cost. Moving to the third sub-neighborhood may be useful as an expensive part is removed and replaced with a less expensive part. Lastly, moving to the fourth sub-neighborhood may be useful to obtain a better stock allocation. It can be shown that the upper bound for the size of the neighborhood defined above is given by $IJ(2+IJ)$.

Having defined the neighborhood structure, we now describe the local search process. Here, we apply a greedy (steepest descent) local search method that allows us to explore the entire neighborhood at each iteration. The method uses an iterative improvement technique. During each iteration, all possible neighbors of the current solution are evaluated, and the one with the minimum total cost is selected. If the obtained new total cost is less than the current total cost, the selected solution becomes the current solution. Otherwise, no local improvement is possible and we take the current solution as the heuristic's solution. Notice that our definition of the neighborhood implies that all non feasible solutions are excluded in the local search process.

As we may have a large neighborhood size in our problem, it is very important to devise an efficient way in the evaluation process of all neighbors. In principal, the approach used in the first initialization algorithm is also used here. To calculate the total cost for each neighbor of the current solution, during the first iteration we need to solve in total $2IJ$ Markov processes for all neighbors in the first and second sub-neighborhoods and $IJ(J-1)$ Markov processes for all neighbors in the fourth sub-neighborhood. For the third sub-neighborhood we can use the results obtained from the first and the second sub-neighborhood. Since any changes of the stock level for an item do not affect the Markov processes of other items, we only need to solve at most $2J(J+1)$ Markov processes for all the next iterations.

Finding the lower bounds

To evaluate the performance of the heuristics, it is useful to compute a lower bound of the optimal total cost. We use the lower bounding procedure based on Lagrangian relaxation as developed in Wong et *al*. (2003). Our evaluation is made based on the distances between the total costs obtained by the heuristics and the lower bounds.

For a given multiplier vector $\lambda \in \Re^J$ with $\lambda_j \geq 0$, $j = 1,\ldots, J$, we formulate the following problem $\boldsymbol{P_1}$ that is obtained from problem $\boldsymbol{P_0}$ by relaxing the waiting time constraints. Notice that the constraints in (5) can be rewritten as

$$\sum\nolimits_{i=1}^{I} m_j^i \left(D_j^i + \theta^i TE^i\right) \leq M_j W_j^{max} \quad j = 1,\ldots, J \tag{9}$$

We then formulate the relaxed problem as follows:

Problem $\boldsymbol{P_1}$:

Minimize
$$Z^{P_1}(\lambda) = \sum\nolimits_{j=1}^{J}\sum\nolimits_{i=1}^{I}\left(CH^i S_j^i + CT^i m_j^i D_j^i + CE^i m_j^i \theta^i\right) + \sum\nolimits_{j=1}^{J}\lambda_j\left(\sum\nolimits_{i=1}^{I} m_j^i (D_j^i + \theta^i TE^i) - M_j W_j^{max}\right) \tag{10}$$

subject to $S_{ij} \in \mathbb{N}_0 \quad i = 1, \ldots, I;\ j = 1, \ldots, J.$ (11)

The original problem $\boldsymbol{P_0}$ is a service model, a model in which the objective is to minimize the average total cost subject to the constraints that certain target service levels have to be met. In our case, the target service levels are represented by the maximum waiting time constraints. By putting the service level constraints in the objective function as in problem $\boldsymbol{P_1}$, we have a so-called cost model, a model without service level constraints. In the cost model, the problem can be decomposed into I independent single-item problems.

It is well known that for each $\lambda \geq 0$, $Z^{P_1}(\lambda)$ represents a lower bound of the optimal cost of problem $\boldsymbol{P_0}$. The sub-gradient optimization method is applied to determine the best Lagrange multipliers that give the tightest lower bound (see Wong et *al*. 2003 for a more detailed description of the lower bounding procedure). This lower bounding procedure can also be considered as a heuristic for the original problem. During the execution of the sub-gradient method, we evaluate the cost for each solution $\underline{S}$ that is feasible in problem $\boldsymbol{P_0}$ and we keep track of the best solution obtained so far. If the final solution of the sub-gradient method is feasible, we may stop. Otherwise, we look for a feasible solution by applying a greedy approach similar to the one used in the first initialization algorithm. The local search algorithm is then applied to further improve the obtained solution.

Computational Experiment

In this section we present and discuss our numerical findings. Our main inquiry will focus on the performance evaluation of the heuristics and the analysis of cost savings obtained by applying a system-approach and pooling policy.

Evaluation of the heuristics

We first describe the set-up of our experiment for evaluating the four different heuristics (the first three heuristics correspond to the three different initialization procedures and the fourth heuristic is the Lagrangian heuristic). Table 1 shows all parameter values used in this experiment. The experiment involves problems with three different numbers of locations (J = 2, 3, and 4) and two different numbers of items (I = 20 and 50). The choice of the values for several input parameters in this experiment is partly based on the data collected from an air carrier company. We focus on a setting with high inventory holding costs in comparison to transportation costs, and a short lateral transshipment and emergency shipment time in comparison to the regular repair lead-time. The ratio of demand rates for each item is generated randomly from a uniform distribution. The value for the repair rates is fixed at $\mu^i = 0.05$/day. The values of the inventory holding cost parameters are generated from two distributions representing two different variabilities of holding cost among items. Further, we examine problems in which all locations have identical and different maximum waiting times.

Due to the integrality of our decision variables, the values of the output parameters may to some extent contain the effect of coincidences. A very small change in one of the input parameters may cause a large change in one of the output parameters. A number of samples could be useful in reducing the effect caused by these coincidences. In this experiment, 10 samples were generated for each combination of I, J, the distribution for generating the maximum waiting times, the distribution for generating the CH^i, and the distribution for generating the demand rates. This gives in total 240 problem sets.

For the evaluation of the heuristics, we computed and recorded the following performance measures:

- $\%GAP$: percentage gap between the total cost obtained by the heuristic and the lower bound

$$\%GAP = \frac{\text{heuristic's total costs - lower bound}}{\text{lower bound}} \text{ x } 100$$

- The number of iterations incurred in the local search procedure
- The computation times required to solve the problem

Table 1. Parameter values for the computational experiments

Name of the parameter	Unit	Number of values	Values
Number of locations (J)		3	2, 3, 4
Number of items (I)		2	20, 50
Inventory holding cost (CH^i)	$/unit/year	2	U[6000,18000], U[3000,21000]
Transshipment cost (CT^i)	$/day	1	1000
Emergency supply cost (CE^i)	$/day	1	1000
Lateral transshipment lead time (TL^i_{jk})	days	1	U[0.15,0.25]
Emergency supply lead time (TE^i)	days	1	1
Maximum waiting time (W^{max}_j)	days	2	0.3, U[0.2,0.4]
Repair rate (μ^i)	/day	1	0.05
Demand rate (m^i_j)	/day	1	U[0.0075,0.1125]

The results of our experiment are summarized in Tables(2)-(4). Tables 2(a)-2(e) present the comparison of *%GAP*. Table 3 presents comparisons of the average number of iterations incurred in the local search procedure for all combinations of I and J. The average computation times of the three heuristics are presented in Table 4 (*the experiment is executed on a PC with a 333-MHz Pentium II processor*).

Table 2. Comparison of *%GAP*

(a) Average *%GAP* with respect to *J*

	$J = 2$	$J = 3$	$J = 4$
Heuristic #1	0.45	0.60	0.78
Heuristic #2	0.68	0.71	0.98
Heuristic #3	2.96	3.01	3.64
Lagrangian	0.32	0.40	0.57

(b) Average *%GAP* with respect to CH^i

	U[6000,18000]	U[3000,21000]
Heuristic #1	0.58	0.64
Heuristic #2	0.81	0.77
Heuristic #3	3.24	3.16
Lagrangian	0.40	0.46

(c) Average *%GAP* with respect to $W_j^{\max}$

	0.3	U[0.2,0.4]
Heuristic #1	0.33	0.89
Heuristic #2	0.68	0.91
Heuristic #3	2.98	3.42
Lagrangian	0.45	0.41

(d) Average *%GAP* with respect to *I*

	$I = 20$	$I = 50$
Heuristic #1	0.72	0.50
Heuristic #2	0.83	0.75
Heuristic #3	3.11	3.29
Lagrangian	0.44	0.42

(e) Overall average *%GAP*

Heuristic #1	0.61
Heuristic #2	0.79
Heuristic #3	3.20
Lagrangian	0.43

Table 3. Average number of iterations in the local search process

	$J = 2$		$J = 3$		$J = 4$	
	I =20	$I = 50$	$I = 20$	$I = 50$	$I = 20$	$I = 50$
Heuristic #1	2.1	6.3	7.6	14.5	7.8	14.8
Heuristic #2	13.8	31.6	24.5	38.9	29.3	42.4
Heuristic #3	11.6	27.0	29.6	53.2	43.0	91.8
Lagrange	2.8	5.8	6.9	12.8	8.6	15.2

Table 4. Average CPU times for the three heuristics (seconds)

	$J = 2$		$J = 3$		$J = 4$	
	I =20	$I = 50$	$I = 20$	$I = 50$	$I = 20$	$I = 50$
Heuristic #1	7	30	38	141	192	533
Heuristic #2	8	38	31	156	234	657
Heuristic #3	14	52	54	238	301	863
Lagrange	73	212	396	858	2177	6121

The main observations drawn from these tables can be summarized as follows:

- Among the first three heuristics, the first heuristic appears to be the best one in terms of the obtained total cost. The quality of the solution obtained by the first heuristic is quite good as indicated by very low *%GAP* with an average of 0.61%. This gap is only a little higher than the gap of the Lagrangian heuristic (0.43%). The second heuristic applying an item approach for the initialization procedure is the second best heuristic with an average *%GAP* of 0.79%. Through our experiment we noted however, that for few instances the second heuristic gave better solutions than the first heuristic. The third heuristic that applies a 'no-pooling' approach for the initialization procedure performs unsatisfactorily and it is dominated by the other two heuristics. It has an average *%GAP* of 3.20%.
- In line with our findings in Wong *et al*,[19] it is shown in Table 2(d) that except for the third heuristic, *%GAP* is decreasing in the number of items, I. Another observation which may be interesting is that the performance of the heuristics is sensitive to the variability of the maximum waiting times at all locations. Table 2(c) shows that all heuristics (especially the first and the second heuristic) perform better when all locations set identical target maximum waiting times.
- The first heuristic is also the most efficient heuristic as its average computation times are the lowest in comparison to the other heuristics (see Table 4). This result is related to the observation that can be drawn from Table 3 where the averages of the number of iterations in the local search process are shown. The first heuristic requires very few iterations in the local search process compared to the other two heuristics. This gives an indication that the greedy-type initialization algorithm provides initial solutions that are close to the (local) optimum. It is also shown that the third heuristic has the longest computation times (among the three first heuristics) although it has the fastest initialization algorithm. This is due the two following reasons. First, the third heuristic starts with initial solutions that have a higher number of stocks in comparison to the first and second heuristic. As a result, the state space of the Markov processes becomes larger and thus longer times are required to obtain numerical solutions of the Markov chains. Second, the third heuristic has also the highest number of iterations in the local search process. This is shown in all problem sets with the exception of the problem sets with J = 2. Another observation is that, as expected, the computation time for the Lagrangian heuristic is considerably high.

The results of this experiment show that the first heuristic applying a greedy-type initialization continued by a local search procedure represents a promising solution procedure for solving multi-item, multi-location spare parts inventory problems. Besides its simplicity, this heuristic is also capable in providing very high quality solutions. As we use an exact model for the evaluation of a stocking policy, we can see in Table 4 that the computation times increase very fast with the number of locations. When dealing with larger problems (e.g. problems with 20 locations), an approximate evaluation procedure such as the one developed in Alfredsson and Verrijdt[8] would be of great help to speed up the computation time of the heuristic.

Analysis of the cost savings

In addition to the evaluation of the heuristics, we are also interested in studying the economic implication of integrating the system approach and pooling policy. More specifically, we are interested in knowing how much money could be saved by applying such policy and which parameters affect the obtained cost savings. For such a purpose, we used all problem sets generated for the evaluation of the heuristics. Additional problem sets are generated to cover problems with a higher number of items (I = 100) and a higher number of pooling members (J = 5 and 6). The analysis is thus based on 600 problem sets. Since the first heuristic is the best heuristic, all calculations in this analysis are based on the total cost obtained by the first heuristic. For each problem set, we computed the following performance measures:

- *%SAVE1* : the percentage cost savings obtained when moving from 'no-pooling and item-approach' strategy to 'pooling and system-approach' strategy.
- *%SAVE2* : the percentage cost savings obtained when moving from 'no-pooling and system-approach' strategy to 'pooling and system-approach' strategy.
- *%SAVE3* : the percentage cost savings obtained when moving from 'pooling and item-approach' strategy to 'pooling and system-approach' strategy.

The results of this experiment are presented in Tables 5(a)-(e). The main observations drawn from these tables can be summarized as follows:

- Applying a system approach together with the pooling policy in making inventory decisions gives quite significant cost savings in comparison to the conventional method that uses an item approach and does not consider pooling. The overall average for *%SAVE1* in this experiment is 39.6% as shown in Table 5(e). It is also shown that the effect of pooling to the total cost improvement is higher than the effect of system approach as the overall average for *%SAVE2* and *%SAVE3* are 24.9% and 9.5% respectively. Further, the average of *%SAVE1* is 5.2% above the sum of the averages of *%SAVE2* and *%SAVE3*. It is important to mention that those magnitude of savings may be representative only for settings with similar parameter values as used in this experiment. The average of *%SAVE3*, for example, that reaches only up to 9.5% in this experiment might be caused by the limited range of inventory holding costs used in the experiment (recall that our experiment allows a maximum ratio of seven between the highest and the lowest inventory holding costs). In a different setting with a single-location model, Thonemann *et al*[3] indicate much higher savings (up to 25%) could be obtained when moving from an item approach to a system approach.

Table 5. Cost savings with respect to different parameters

(a) Percentage savings with respect to J

	$J = 2$	$J = 3$	$J = 4$	$J = 5$	$J = 6$
%SAVE1	31.8	36.7	40.5	43.1	45.4
%SAVE2	16.0	21.1	25.6	29.2	32.6
%SAVE3	10.4	9.9	9.6	8.8	8.8

(b) Percentage savings with respect to $W_j^{\max}$

	0.3	U[0.2,0.4]
%SAVE1	41.9	37.3
%SAVE2	26.7	23.1
%SAVE3	10.2	8.8

(c) Percentage savings with respect to CH^i

	U[6000,18000]	U[3000,21000]
%SAVE1	38.8	40.4
%SAVE2	25.0	24.8
%SAVE3	8.3	10.7

(d) Percentage savings with respect to I

	$I = 20$	$I = 50$	$I = 100$
%SAVE1	38.4	38.9	41.5
%SAVE2	23.2	25.6	25.9
%SAVE3	9.2	9.6	9.7

(e) Overall percentage savings

%SAVE1	39.6
%SAVE2	24.9
%SAVE3	9.5

- Both *%SAVE1* and *%SAVE2* are increasing with the number of cooperating companies, *J*, but the increase rate is slowing down. *%SAVE3* is decreasing with *J* although the decrease rate is not so large.
- The variability of inventory holding cost among items has some impacts on *%SAVE1* and *%SAVE3* as indicated in Table 5(c). The percentage cost savings increase when the

variability is higher. This result is also in line with the findings of Thonemann *et al*[3] Wong *et al*.[19]

- Table 5(b) shows that the three types of savings are higher when the maximum waiting times are identical. This result however, should be interpreted with more caution as it is known from Table 4(c) that all heuristics also perform better when the maximum waiting times are identical for all locations (for the first heuristic, the difference of *%GAP* is 0.56%). However, smaller differences in *%SAVE3* compared to differences in *%SAVE1* and *%SAVE2* may indicate that both *%SAVE1* and *%SAVE2* are indeed sensitive to the variability of the maximum waiting times.

Conclusions and directions for further research

In this paper we have developed a solution procedure for solving the optimization problem in multi-item spare parts systems where lateral and emergency shipments can occur in response to stock-outs and there exist constraints of the target maximum waiting times for a ready-for-use at all locations that have to be satisfied. We structured the optimization problem as a combinatorial problem and developed a solution procedure based on a local search optimization method. The solution procedure mainly consists of two steps: initialization and improvement. Three different initialization algorithms were formulated and a steepest-descent local search method has been used for the improvement step. A computational experiment was performed to evaluate the relative merits of the three heuristics. The Lagrangian relaxation based approach is used to obtain the lower bounds of the optimal total costs and the evaluation of heuristic is made based on the relative distance between the heuristic solutions and the lower bounds. The results of the experiment show that the heuristic applying a greedy-type method for the initialization is the best one and its performance is quite good as its total costs have an average distance of 0.61% to the lower bounds. In terms of the computation time, this heuristic is also the most efficient one. Compared to the Lagrangian heuristic developed in Wong *et al*,[19] the new heuristic gives quite a significant reduction in computation times with nearly the same quality.

In addition to the evaluation of the heuristics, our numerical experiment has also shown that significant benefits are obtained through spare parts pooling and the application of a system approach. More specific conclusions are summarized as follows: (1) the relative cost savings of applying a pooling policy are higher than the relative cost savings of applying a system approach; (2) the relative cost savings of applying a pooling policy increase with the number of pooling members with a decreasing increase rate; (3) the relative cost savings of applying a pooling policy are higher when all pooling members set identical maximum waiting times; (4) the relative cost savings of applying a system approach increase when the variability of inventory holding costs among items increases.

Our work can be extended in several directions. One possible extension is to consider a two-echelon setting. This extension is relevant since there is a trend nowadays towards outsourcing the *MRO* operations (Maintenance, Repair and Overhaul). Consequently, pooling will move more into a vendor or third-party model where a neutral independent company (at the first echelon) will offer component pooling options to companies (at the second echelon). The presence of a pooling provider company at the first echelon complicates the problem as the logistical performances of the companies in the pool depend not only on the demand rate and the stocking levels at their locations, but also on the stocking levels of the pooling provider.

References

1 Kennedy WJ, Patterson JW and Fredendall LD (2002). An overview of recent literature on spare parts inventories. *Int J Prod Econs* **76**: 201-215.

2 Sherbrooke CC (1992). *Optimal Inventory Modeling of Systems*. John Wiley & Sons, New York.

3 Thonemann UW, Brown AO and Hausmann WH (2002). Easy quantification of improved spare parts inventory policies. *Mngt Sci* **48**: 1213-1225.

4 Rustenburg JW, Van Houtum GJ and Zijm WHM (2003). Exact and approximate analysis of multi-echelon, multi-indenture spare parts systems with commonality. In: Shantikumar JG, Yao DD and Zijm WHM (eds). *Stochastic Modeling and Optimization of Manufacturing Systems and Supply Chains*. Kluwer, Boston.

5 Lee HL (1987). A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Mngt Sci* **33**: 1302-1316.

6 Axsäter S (1990). Modeling emergency lateral transshipments in inventory systems. *Mngt Sci* **36**: 1329-1338.

7 Sherbrooke CC (1992). Multi-echelon inventory systems with lateral supply. *Naval Res Log* **39**: 29-40.

8 Alfredsson P and Verrijdt J (1999). Modeling emergency supply flexibility in a two-echelon inventory system. *Mngt Sci* **45**: 1416-1431.

9 Kukreja A, Schmidt CP and Miller DM (2001). Stocking decisions for low-usage items in a multi-location inventory system. *Mngt Sci* 47: 1371-1383.

10 Gross D (1963). Centralized inventory control in multi-location supply systems. In Scarf HE, Gilford DM and Shelly MW (eds). *Multistage Inventory Models and Techniques*. Stanford University Press, Stanford.

11 Krishnan KS and Rao VRK (1965). Inventory control in N warehouses. *J Ind Eng* **16**: 212-215.

12 Das C (1975). Supply and redistribution rules for two-location inventory systems: one-period analysis. *Mngt Sci* **21**: 765-776.

13 Karmarkar US (1987). The multi-location multi-period inventory problem: bounds and approximations. *Mngt Sci* **33**: 86-94.

14 Tagaras G (1989). Effects of pooling on the optimization and service levels of two-location inventory systems. *IIE Trans* **21**: 250-257.

15 Tagaras G (1999). Pooling in multi-location periodic inventory distribution systems. *OMEGA, Int J Mgt Sci* **27**: 39-59.

16 Robinson LW (1990). Optimal and approximate policies in multi-period, multi-location inventory models with transshipments. *Opns Res* **38**: 278-295.

17 Tagaras G and Cohen MA (1992). Pooling in two-location inventory systems with non-negligible replenishment lead times. *Mngt Sci* **38**: 1067-1083.

18 Herer YT and Rashit A (1999). Lateral stock transshipments in a two-location inventory system with fixed and joint replenishment costs. *Naval Res Log* 46: 525-547.

19 Wong H, Van Houtum GJ, Cattrysse D and Van Oudheusden D. *Multi-item spare parts systems with lateral transshipments and waiting time constraints*. Working Paper, 2003, Centre for Industrial Management, Katholieke Universiteit Leuven.

20 Archibald TW, Sassen SA and Thomas LC (1997). An optimal policy for a two depot inventory problem with stock transfer. *Mngt Sci* **43**: 173-183.

21 Wong H. *Inventory pooling of repairable spare parts: models for operational and financial decisions*. PhD Thesis, 2004, Centre for Industrial Management, Katholieke Universiteit Leuven.