An approximate approach for the joint problem of level of repair analysis and spare parts stocking

Beta
Research School for Operations
Management and Logistics

An approximate approach for the joint problem of level of repair analysis and spare parts stocking

R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten

Beta Working Paper series 347

BETA publicatie	WP 347 (working paper)
ISBN	978-90-386-2496-9
ISSN	
NUR	804
Eindhoven	April 2011

An approximate approach for the joint problem of level of repair analysis and spare parts stocking

R.J.I. Basten • M.C. van der Heijden • J.M.J. Schutten

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

University of Twente, School of Management and Governance
P.O. Box 217, 7500 AE, Enschede, The Netherlands

April 14, 2011

Abstract

For the spare parts stocking problem, generally METRIC type methods are used in the context of capital goods. Implicitly, a decision is assumed on which components to discard and which to repair upon failure, and where to perform repairs. In the military world, this decision is taken explicitly using the level of repair analysis (LORA). Since the LORA does not consider the availability of the installed base, solving the LORA and spare parts stocking problems sequentially may lead to suboptimal solutions. We propose an iterative algorithm to solve the two problems. We compare its performance with that of the sequential approach and a recently proposed, so-called integrated algorithm. The latter finds optimal solutions for two-echelon, single-indenture problems. In our experiment, we use a set of such problems, and a set of multi-echelon, multi-indenture problems, for which we achieve a cost reduction of 3% on average (35% at maximum) compared with the sequential approach. Compared with the integrated algorithm, the gap is only 0.6% on average (5% at maximum), while the maximum computation time falls from 3 hours to 2.5 minutes. In a case study, we achieve a cost reduction of 10% compared with the sequential approach.

Service logistics, Level of repair analysis, Spare parts, Inventories

1 Introduction

Capital goods are physical systems that are used to produce products or services. They are expensive and technically complex, and they have high downtime costs. Examples of capital goods are manufacturing equipment, defense systems, and medical devices. Before capital goods are deployed, several tactical level questions concerning their corrective maintenance need to be answered: which components to repair upon failure and which to discard, where to perform the repairs, and which amount of spare parts to stock at which locations in the repair network. These questions should be answered such that a target availability of the installed base is achieved against the lowest possible costs.

Due to the high downtime (unavailability) costs of capital goods, a defective component will usually be *repaired by replacement* of a spare part. In the defense industry, the components that are replaced are called *LRUs* or *line replaceable units.* Defective LRUs may be discarded, which means that a new one needs to be acquired. However, since LRUs are often expensive, they are generally repaired, typically by replacement of a subcomponent, called *SRU* or *shop replaceable unit.*

Figure 1 gives an example of a *multi-indenture product structure*, including the naming convention that we use. We use the terms *component* and *subcomponent* if the indenture level is irrelevant. Repairs and discards can be performed at various echelon levels in the *multi-echelon repair network*, an example of which is shown in Figure 2. We use the naming convention as depicted in this figure. To be able to perform repairs, discards, or movements of components, *resources* may be required. Resources include test, repair, and transportation equipment, but one time training of service engineers may also be considered as a resource for which a one-time investment is required. Spare parts need to be located in the repair network as well, in order to enable quick repairs of both the installed base (spare LRUs) and of LRUs and SRUs (spare SRUs and parts, respectively).

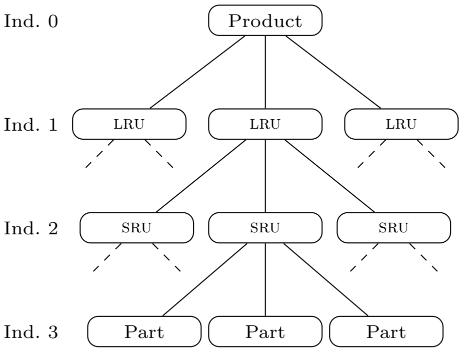

Figure 1: Product structure

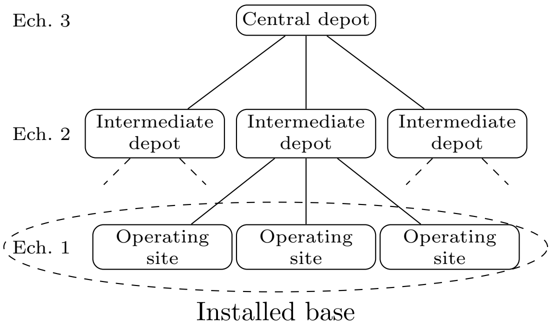

Figure 2: Repair network

The *level of repair analysis* (*LORA*) can be used to make the above mentioned *repair/-discard decisions*:

1. which components to repair upon failure and which to discard; and
2. at which locations in the repair network to perform the repairs and discards.

The third set of decisions that is made in the LORA is:

3. at which locations to deploy resources.

The LORA is typically modeled as a deterministic integer linear optimization problem (see, e.g., Basten et al. 2009, 2010). This means that it is *not* possible to model the downtime waiting for spares and the availability of the installed base, since that would result in a non-linear constraint. Since the spare parts costs can be included only in a rudimentary way, usually only the other relevant costs are considered, consisting of both fixed costs and costs that are variable in the number of failures. Fixed costs are due to the resources. They result from a certain repair/discard decision, but are incurred no matter how often components are actually repaired or discarded, for example, costs for training of service engineers and depreciation costs of repair equipment. Variable costs may include transportation costs, working hours of service engineers, and usage of bulk items.

Using the results of the LORA as an input, the *spare parts stocking* analysis is conducted to decide which components to put on stock at which location(s) in the repair network in which quantity, such that a target availability of the capital goods is achieved against minimum spare parts costs. A well-known method to conduct this analysis is (VARI-)METRIC (see, e.g., Sherbrooke 2004, Muckstadt 2005).

Performing the LORA first and next the spare parts stocking analysis, the *sequential approach*, may lead to a solution that is not optimal. For example, if repairs are performed at the operating sites, each operating site requires a resource, whereas only one resource is required in total if repairs are performed at the central depot. As a result, the LORA often recommends to perform repairs at the central depot (repairing centrally leads to higher transportation costs of components, but these costs are generally low compared to costs for resources in a high-tech environment). The LORA neglects the fact that when repairs are performed centrally, the repair lead times (including transportation lead times) are higher than when repairs are performed at the operating sites, thus leading to higher spare parts requirements. This is especially problematic if the spare parts costs make up a large percentage of the total costs, as we have observed in a case study in the defense industry (see Section 6).

We propose an iterative algorithm to solve the joint problem of LORA and spare parts stocks. The basic idea is that we use the results of the spare parts stocking problem to adapt the LORA inputs. So, we solve a LORA first and then use VARI-METRIC to solve the spare parts stocking problem. We next adapt the LORA inputs to start a second iteration and continue in this way until we do not find a different solution anymore. We compare our results with the sequential solution (this is the solution of the first iteration of our algorithm) and with the solutions resulting from an algorithm that was recently proposed by Basten et al. (2011a) for two-echelon, single-indenture problems. Their so-called *integrated algorithm* finds optimal solutions, or in fact, efficient points on the curve of costs versus expected number of backorders (see Section 2.2). The integrated algorithm can be extended to multi-echelon, multi-indenture problems, but Basten et al. (2011a, Appendix) explain that in that case, finding efficient points cannot be guaranteed. However, we believe that the integrated algorithm still finds solutions that are close to optimal (see also Section 5). The major drawback of the integrated algorithm is that it is very slow due to its enumerative approach.

We perform an extensive numerical experiment. On a set of two-echelon, single-indenture problems, the iterative algorithm achieves a cost reduction of 3.80% on average compared with the sequential approach, whereas the integrated approach achieves a cost reduction of 5.07% on average. This means that the iterative algorithm closes most of the optimality gap of the sequential approach. Using a set of multi-echelon, multi-indenture problems, we find that the iterative algorithm is much faster than the integrated algorithm, while its solution value is on average only 0.58% higher than that of the integrated algorithm (5.26% higher at maximum). Compared with the sequential procedure, the iterative algorithm achieves a cost reduction of 2.85% on average and 34.69% at maximum. In a case study at Thales Nederland, a manufacturer of naval sensors and naval command and control systems, we show that solving the joint problem iteratively instead of sequentially leads to a cost reduction of almost 10% which is worth millions over the life time of a couple of sensor systems.

The remainder of this paper is organized as follows. In Section 2, we discuss the related literature. We outline our model for the joint problem of LORA and spare parts stocking in Section 3, and in Section 4, we present the iterative algorithm. In Section 5, we show the results of our numerical experiment, and we then present the results of the case study that we performed in Section 6. We give conclusions and recommendations for further research in Section 7.

2 Literature review

We discuss the literature on LORA, spare parts stocking, and the joint problem of LORA and spare parts stocking in Sections 2.1, 2.2, and 2.3, respectively.

2.1 Level of repair analysis

Barros (1998) proposes a multi-echelon, multi-indenture LORA model in which decisions are taken per echelon level. So, if it is decided to repair a certain component at a certain operating site, it is also repaired at all other operating sites. Barros further assumes that all components at a certain indenture level require the same resource and that resources are uncapacitated. The latter means that there is no downtime waiting for resources, and either zero or one resource is located at each location. As in all papers on LORA, Barros formulates her model as an integer programming model. She solves it using CPLEX. Barros and Riley (2001) use the same model as Barros does and solve it using a branch-and-bound approach.

Saranga and Dinesh Kumar (2006) make the same assumptions as Barros (1998), except that the former assume that each component requires its own unique resource. They use a genetic algorithm to solve the model. Basten et al. (2009) generalize the two aforementioned models by allowing for components requiring multiple resources and multiple components requiring the same resource. As in the remaining three papers in this section, Basten et al. (2009) use CPLEX to solve the model.

Basten et al. (2010) generalize the model of Basten et al. (2009) by allowing for different decisions at the various locations at one echelon level. They show that the LORA problem can be modeled efficiently as a generalized minimum cost flow model. Basten et al. (2011b) propose a number of extensions to the model of Basten et al. (2010) so that, for example, a probability of unsuccessful repair can be modeled, or capacitated resources. The latter does not mean that waiting times are incorporated.

Brick and Uchoa (2009) use similar assumptions as Basten et al. (2010), except that the former assume that resources have a maximum capacity (as Basten et al. 2011b, do). They further consider one echelon level only and effectively assume two indenture levels. Integrated in their LORA is the decision of which facilities to open (facility location problem).

2.2 Spare parts stocking

In the area of capital goods, the paper of Sherbrooke (1968) is generally seen as the seminal paper on the multi-item spare parts stocking problem. Sherbrooke develops the METRIC model (Multi-Echelon Technique for Recoverable Item Control), which is the basis for a huge stream of METRIC type models. The goal in these models is to find the most cost effective allocation of spare parts in a network, such that a target availability of the installed base is achieved. This is achieved by focusing on the expected number of backorders: if a spare part is requested, but not available yet, this is called a *backorder*. As an approximation, the number of backorders of LRUs at operating sites equals the number of systems that are unavailable waiting for spares. The METRIC type methods focus on minimizing the expected number of backorders, instead of maximizing the availability, because this allows for decomposing the overall problem into subproblems per LRU. A marginal analysis approach is used to construct an EBO-curve of spare parts costs versus expected number of backorders. Construction of the curve is stopped as soon as the number of backorders has decreased enough to achieve the target availability. Generally, the achieved availability is somewhat higher than the target availability, which is called *overshoot*.

A key assumption in these models is that demand at the operating sites follows a Poisson process, which means that demand at higher echelon levels follows a Poisson process as well. As a result, the number of components in repair or in the replenishment loop (after discard) at the highest location is Poisson distributed. However, the number of backorders at that location is *not* Poisson distributed if spare parts are located there. As a result, analysis of the so-called pipeline at the lower echelon levels gets complicated, the pipeline being the number of components that is sent upwards for repair or discard, and not replaced by a functioning component yet. Sherbrooke (1968) chooses to approximate the number of items in the pipeline by assuming that it is also Poisson distributed. Muckstadt (1973) extends the work by Sherbrooke (1968); the latter considers single-indenture product structures only, whereas the former develops a multi-echelon, multi-indenture model, called MOD-METRIC. The development of the VARI-METRIC models (Graves 1985, Sherbrooke 1986) has been the next important step forward: a two-moment approximation is used for the pipelines. It is also possible to evaluate the model exactly (Graves 1985, Rustenburg et al. 2003), but this is computationally intensive, and VARI-METRIC is known to give small errors only. Furthermore, backorders at higher echelon levels are not the only cause of delays; backorders for subcomponents can delay repairs of components in a way that is similar to what we described above.

2.3 Joint problem of level of repair analysis and spare parts stocking

We are aware of two papers in which a method is presented to solve the joint problem of LORA and spare parts stocking: Alfredsson (1997) and Basten et al. (2011a).

Alfredsson (1997) assumes a single-indenture product structure and a two-echelon repair network. He further assumes that each component requires exactly one tester (resource) and that all components that require the same tester are repaired at the same location. Furthermore, one multi-tester exists. It can be used for the repair of a number of components, and adapters can be added in a fixed order to enable the multi-tester to be used for the repair of additional components. If the multi-tester can be used to repair a certain component, then this component necessarily uses the multi-tester instead of the original resource that it used. Resources are capacitated, which means that multiple resources of the same type may be required at one location. System downtime includes the waiting times for the resources, the repair times, and the waiting times for spares. The problem is modeled as a non-linear integer programming model and Alfredsson uses a decomposition method that sequentially decomposes the overall problem into smaller subproblems to solve the model.

Basten et al. (2011a) also consider single-indenture, two-echelon problems, but they allow for more general component-resource relations: components may share resources and a component may require multiple resources simultaneously. This substantially complicates the problem. The basic idea is to recursively decompose the problem in a smart way such that all possible solutions are enumerated without taking too much time. For the single-indenture, two-echelon problem, Basten et al. (2011a) find EBO-curves consisting of efficient points. This means that it is not possible to achieve a lower expected number of backorders for the costs that they find. They also show that it cannot be guaranteed that efficient points are found for general multi-indenture problems, but that solutions are probably still close to optimal.

3 Model

In Section 3.1, we give the assumptions that we make, and we present the mathematical model formulation in Section 3.2.

3.1 Assumptions

A key assumptions is that we take the same decisions at all locations at one echelon level for each component and resource. This implicitly means that we assume symmetrical repair networks, i.e., we have the same costs, demand rates, lead times, et cetera at all locations at one echelon level, and the same number of locations attached to a location at the higher echelon level. In such a network, taking the same decision at all locations at one echelon level is an optimal strategy, except that the overshoot increases (see Section 2.2).

Next, we make some assumptions that are commonly made for the LORA problem:

- discarding a component implies that its subcomponents are discarded as well;
- each subcomponent may be the cause of a failure of a component (otherwise, it need not be modeled), so repairing a component may result in replacement of any one subcomponent. As a result, if it is decided to repair a component at a certain echelon level, a decision needs to be taken for each subcomponent at the same echelon level;
- a failed (sub)component may not be shipped to a lower echelon level;

Furthermore, we make a few assumptions that are standard in the METRIC type models.

- components fail according to a Poisson process with a constant rate;
- a failure in a component is caused by a failure in at most one subcomponent;
- replacement of a failed component by a functioning component takes zero time;
- resources are uncapacitated, meaning that at most one resource of a certain type needs to be installed at each location;
- minimizing the expected number of backorders is a good approximation of maximizing the availability (see Section 2.2);
- there are no lateral transshipments between locations at the same echelon level or emergency shipments from locations at a higher echelon level; functioning spare parts are only supplied from one specific location at the next higher echelon level;
- for each component at each location, an $(S-1, S)$ continuous review inventory control policy (one for one replenishment) is used;
- the repair lead times are i.i.d. random variables that include the time used for sending the failed component to the repair location and for diagnosing the failure cause;
- the replenishment lead times for discarded components are i.i.d. random variables that include the time used for diagnosing the failure cause;

- the move lead time (to move a functioning, repaired or newly purchased, component from a location to one of its child locations) is deterministic;
- repairs are always successful.

To ease the presentation in the remainder of this paper and to decrease the problem size, we make three additional assumptions that are not critical for our algorithm:

- there is no commonality, so a subcomponent may not be part of two different components;
- since resources that are required to enable discard or movement do not occur frequently in practice, e.g., not in our case study, we assume that resources may be required to enable repair only;
- since the discard costs mainly consist of the costs of acquiring a new component, we assume that discard costs for a certain component are equal at all echelon levels. Since the costs to move a component to another echelon level are relatively low, we consider discard at the highest echelon level only.

3.2 Mathematical model

In Section 3.2.1, we introduce the notation that we use and we give the mathematical model formulation in Section 3.2.2.

3.2.1 Notation.

Let C be the set of all components, with $C_1 \subseteq C$ being the set of LRUs. Γ_c is the (possibly empty) set of subcomponents of component $c \in C$ at the next higher indenture level.

The set E consists of all echelon levels, the highest echelon level being e_{MAX}. The set D consists of the possible decisions that can be made: $D = \{\text{discard}, \text{repair}, \text{move}\}$. The set of options that is available at echelon level $e \in E$ is D_e. For $e \in E \setminus e_{\text{MAX}}$, $D_e = D$, and $D_{e_{\text{MAX}}} = \{\text{discard}, \text{repair}\}$.

Let R be the set of resources. $\Omega_r \subseteq C$ is the set of components that require resource r in order to be repaired.

We define the following decisions variables:

$$X_{c,e,d} = \begin{cases} 1, & \text{if for component } c \in C \text{ at echelon level } e \in E \text{ decision } d \in D_e \text{ is made,} \\ 0, & \text{otherwise;} \end{cases}$$

$$Y_{r,e} = \begin{cases} 1, & \text{if resource } r \text{ is located at echelon level } e, \\ 0, & \text{otherwise;} \end{cases}$$

$$S_{c,e} = \text{the number of spare parts of component } c \text{ located at each location at echelon level } e.$$

Furthermore, we denote $\mathcal{X}$ as the three-dimensional array with entries $X_{c,e,d}$ and $\mathcal{S}$ as the two-dimensional array with entries $S_{c,e}$.

For each component $c \in C$, we define λ_c (> 0) as the total annual failure rate over all operating sites. We define three cost types. For component $c \in C$ at echelon level $e \in E$, $vc_{c,e,d}$ (≥ 0) are the variable costs of making decision $d \in D$. Since we have chosen, without loss of generality, to minimize the total annual costs with our definition of λ_c, we define $fc_{r,e}$ (≥ 0) to be the annual fixed costs to locate resource $r \in R$ at echelon level $e \in E$ and we define $hc'_{c,e}$ (> 0) to be the annual costs of holding one spare of component $c \in C$ at each location at echelon level e (we use the prime to ease notation later on).

3.2.2 Mathematical model formulation.

We define our model as follows:

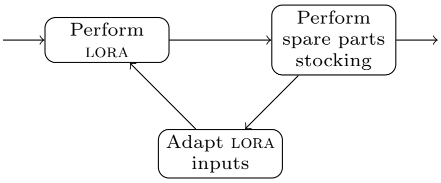

Figure 3: Iterative algorithm

$$\text{minimize} \sum_{c\in C}\sum_{e\in E}\sum_{d\in D_e} vc_{c,e,d}\cdot\lambda_c\cdot X_{c,e,d} + \sum_{r\in R}\sum_{e\in E} fc_{r,e}\cdot Y_{r,e} + \sum_{c\in C}\sum_{e\in E} hc'_{c,e}\cdot S_{c,e} \tag{1}$$

subject to:

$$\sum_{d\in D_1} X_{c,1,d} = 1 \ , \forall c \in C_1 \tag{2}$$

$$X_{c,e,\text{move}} \leq \sum_{d\in D_{e+1}} X_{c,e+1,d} \ , \forall c \in C \ , \forall e \in E \setminus e_{\text{MAX}} \tag{3}$$

$$X_{c,e,\text{repair}} \leq \sum_{d\in D_e} X_{b,e,d} \ , \forall c \in C \ , \forall b \in \Gamma_c \ , \forall e \in E \tag{4}$$

$$X_{c,e,\text{repair}} \leq Y_{r,e} \ , \forall r \in R, \forall c \in \Omega_r \ , \forall e \in E \tag{5}$$

$$\text{availability}(\mathcal{X},\mathcal{S}) \geq \text{target availability} \tag{6}$$

$$X_{c,e,d}, Y_{r,e} \in \{0,1\} \tag{7}$$

$$S_{c,e} \in \mathbb{N} \tag{8}$$

Constraints 2 to 5 are the 'LORA constraints' and define the same model as Basten et al. (2010) use, except that they do not necessarily take the same decision at each location at one echelon level. Constraint 2 assures that for each LRU a decision is made at the operating site. If a component is discarded, no further decisions need to be made for that component or its subcomponents. If a component is moved, Constraint 3 assures that a decision is made for that component at the next higher echelon level, and if a component is repaired, Constraint 4 assures that a decision is made for each of its subcomponents. Some options are only available if all resources are present, which is guaranteed by Constraint 5. Finally, Constraint 6 assures that the target availability is met; this is the only 'spare parts stocking constraint'. Since the availability is a non-linear function of all repair/discard decisions $\mathcal{X}$ and all spare parts decisions $\mathcal{S}$, our model cannot be solved using an ILP solver. Notice that in Constraint 6, the various lead times play a role (we have not introduced notation for those lead times).

4 Iterative algorithm

As mentioned in Section 1, the joint problem of LORA and spare parts stocking analysis is in practice usually solved sequentially. First, a LORA is performed, focusing on achieving the lowest possible costs, consisting of both fixed costs ($\sum_{r\in R}\sum_{e\in E} fc_{r,e}\cdot Y_{r,e}$), and costs that vary with the number of failures ($\sum_{c\in C}\sum_{e\in E}\sum_{d\in D} vc_{c,e,d}\cdot\lambda_c\cdot X_{c,e,d}$). Next, given the decisions that result from the LORA, a spare parts stocking problem is solved (e.g., using VARI-METRIC) that determines where to locate spare parts in the repair network, such that a target availability of the installed base is achieved against the lowest possible spare parts holding costs ($\sum_{c\in C}\sum_{e\in E} hc'_{c,e}\cdot S_{c,e}$).

We propose an iterative algorithm that uses in each iteration the same two building blocks as the sequential approach (see figure 3). After the first iteration, we therefore have the solution of the sequential method. The spare parts holding costs are then used to adapt the

			Spare parts holding costs ($hc^{i+1}_{c,e,d}$) after					
	LORA costs		iteration 1		iteration 2		iteration 3	
	Component		Component		Component		Component	
Decision	A	B	A	B	A	B	A	B
Repair at ship	32	62	0	0	0	0	***4***	0
Repair at depot	22	37	***16***	0	16	***20***	16	***15***
Discard	30	30	0	***30***	***20***	30	20	30
Total costs (LORA and spares)			98		107		88	

Table 1: Costs in the LORA problem (×€1,000)

LORA inputs so that a second iteration of LORA and spare parts stocking may be performed. The key idea is that, as an approximation, we may decompose the spare parts holding costs into spare parts holding costs per component, so that for each component c the spare parts holding costs are $\sum_{e \in E} hc'_{c,e} \cdot S_{c,e}$. The implicit assumption is that the spare parts holding costs that result from a repair/discard decision are independent of the decisions taken for the other components. Of course, this assumption is violated, since VARI-METRIC is a multi-item approach (an example is given below).

Notice that the move decisions can be seen as 'intermediate' decisions; the decision to repair or discard a component is the 'final' decision. Therefore, we need to adapt the costs for the repair and discard decisions only. We define $hc^i_{c,e,d}$ as the spare parts holding costs that are added to the variable costs of component $c \in C$ for decision $d \in \{\text{discard}, \text{repair}\}$ at echelon level $e \in E$ in iteration $i \geq 1$. So, the variable costs that are used in the LORA for component $c \in C$ at echelon level $e \in E$ in iteration i are $vc_{c,e,d} + hc^i_{c,e,d}$ for decision $d \in \{\text{repair}, \text{discard}\}$, and $vc_{c,e,d}$ for decision $d = \text{move}$. In the first iteration, $hc^1_{c,e,d} = 0$ for all tuples (c, e, d). For each tuple (c, e, d) for which $X_{c,e,d} > 0$ in iteration $i-1$ $(i > 1)$, we set $hc^i_{c,e,d} = \frac{\sum_{f \in E} hc'_{c,f} \cdot S_{c,f}}{\lambda_c}$ ($S_{c,f}$ resulting from iteration $i-1$). For all other possible repair and discard decisions ($X_{c,e,d} = 0$ in iteration $i-1$), we set $hc^i_{c,e,d} = hc^{i-1}_{c,e,d}$. This means that the spare parts holding costs that we include in the LORA inputs are changed in iteration i only if the related repair/discard decision was chosen in iteration $i-1$. In this way, we gradually find an estimate for the resulting spare parts holding costs for all relevant repair/discard decisions and the algorithm will eventually find a LORA solution that leads to low *total* costs: LORA costs (excluding the added spare parts holding costs) plus spare parts holding costs resulting from the spare parts stocking analysis. We stop the algorithm as soon as the LORA solution is identical in two consecutive iterations. If in the second of these iterations, two different LORA solutions exist that lead to the same costs, we choose the one we also had in the previous iteration so that the algorithm terminates. In Appendix B, we show that the algorithm cannot cycle between two solutions and that it therefore necessarily terminates after a finite number of iterations.

We use an example to illustrate the feedback mechanism. We consider a radar system that consists of two components (A and B). The radar system is installed at two ships, which are supported by a depot. Components A and B both require a unique resource in order to enable repair, the fixed annual costs of which are €10,000 and €25,000, respectively. For both components, the annual failure rate per ship is 1, the discard costs are €15,000, the variable repair costs are €6,000, and the move costs are €0.

In the first iteration, spare parts holding costs of zero are included in the LORA problem. Therefore, the repair/discard options with the lowest LORA costs are chosen for both components (see Table 1 for an overview of all resulting costs): A is repaired at depot, which leads to annual costs of €22,000 (variable repair costs are €12,000, since there are two failures, one at each ship, and a resource at the depot costs €10,000), and B is discarded, which leads to annual discard costs of €30,000. Next, the spare parts stocking problem is solved, which results in stocking spare parts at both the ships and the depot, leading to annual spare parts holding costs of €16,000 for A and €30,000 for B. In the second iteration, the LORA is solved with modified inputs. The LORA chooses to discard A, since that leads to costs of €30,000, whereas repair at depot leads to total costs of €22,000 + €16,000 = €38,000. For B, repair at depot is the most cost effective option. We next find spare parts holding costs of €20,000 for both components. Notice that the total costs in the second iteration (€107,000) are higher than those in the first iteration (€98,000). In the third iteration, it is decided to repair A at ship, and B at depot. This results in annual spare parts holding costs of €4,000 for A and €15,000 for B.

Notice that the spare parts holding costs for B change, although the repair/discard decision for B does not change. This is a result of the system approach that is used in VARI-METRIC. We simply replace the old costs by the newly calculated costs. Notice furthermore that for A, we found the spare parts holding costs estimate related to 'repair at depot' when 'discard' was chosen for component B. This value may be lower if B is repaired at ship or at depot and as a result, in the optimal solution, we may have to repair A at depot. However, the solution in the next iterations will be to repair A at ship and to repair B at depot. This risk of using spare parts holding costs that are too high is the key drawback of our approach and it may result in not selecting a cost-effective option anymore. As a result, we may end up with a non-optimal solution.

It is possible to slightly improve the feedback algorithm. For example, instead of replacing an old value ($hc^{i-1}_{c,e,d}$) by a new value ($hc^{i}_{c,e,d} = \frac{\sum_{f \in E} hc'_{c,f} \cdot S_{c,f}}{\lambda_c}$), we may take a weighted average of the old and new value ($hc^{i}_{c,e,d} = \alpha \cdot \frac{\sum_{f \in E} hc'_{c,f} \cdot S_{c,f}}{\lambda_c} + (1-\alpha) hc^{i-1}_{c,e,d}$, with $0 < \alpha < 1$). However, such improvements require setting additional values (what is a good value for α?), they make the algorithm more difficult to grasp and implement, and they lead to higher computation times. Therefore, we do not consider them here. In his Ph.D. thesis, Basten (2010) shows the results of implementing three such improvements.

5 Numerical experiment

We design a numerical experiment that we present in Section 5.1. In Section 5.2, we discuss the results of our tests by answering the following questions:

1. What cost reduction can be achieved by solving the joint problem of LORA and spare parts stocking iteratively instead of sequentially?
2. How does the iterative algorithm perform compared with the integrated algorithm?
3. Which model parameters influence the cost reductions that may be achieved by solving the joint problem using the integrated or iterative algorithm instead of sequentially?
4. How do the repair strategies change when solving the joint problem using the integrated or iterative algorithm instead of sequentially?

For two-echelon, single-indenture problem instances, Question 2 effectively means comparison of the iterative algorithm with the optimal solution. Therefore, our experiment consists of a set of two-echelon, single-indenture problem instances, and a set of multi-echelon, multi-indenture problem instances.

The algorithms are implemented in Delphi 2007 and problems instances are solved on an Intel Core 2 Duo P8600@2.40 GHz, with 3.5 GB RAM, under Windows XP SP 3. For the iterative and sequential algorithm, the LORA building block consists of the model of Basten et al. (2009) (but implemented using the minimum cost flow model of Basten et al. 2010). The spare parts stocking building block in the sequential algorithm consists of either the exact or approximate, two-moment evaluation of Graves (1985), with the optimization as described by Muckstadt (2005) and Basten et al. (2011a) (which is exact if exact evaluation is used). The approximate evaluation is extended to the general multi-echelon, multi-indenture problem as described in Rustenburg et al. (2003). In the iterative algorithm, the approximate evaluation is always used. The integrated algorithm is an extension of the work by Basten et al. (2011a). As in the sequential approach, we use either the exact or approximate evaluation of Graves (1985). We clearly indicate which one is used in each test.

	Default	Additional	Varied in
Parameter	value	value(s)	test set
# Echelon levels	3	2	1
# Central depots	1	—	—
# Intermediate depots	2	—	—
# Operating sites[1]	10	—	—
# Indenture levels	3	2	1
# LRUs	50	100	1
# Subcomponents per parent	2	—	—
# Resources	10	—	—
# Component types	4	3	3
% Resources used by 1 component	50%	0%	3

Table 2: Fixed values in the experiments

5.1 Design

First, we explain our main problem instances, which are multi-echelon, multi-indenture problems instances, in Section 5.1.1. Next, we discuss in Section 5.1.2 the problem instances that Basten et al. (2011a) have used in their numerical experiment.

5.1.1 Multi-echelon, multi-indenture problem instances

We use three sets of problem instances, each having its own focus:

1. Varying the problem size, the spare parts holding costs, and the lead times.
2. Varying the attractiveness of acquiring resources by changing the annual number of failures and the costs of resources and components (resulting in different variable repair, discard, and move costs).
3. Varying the component-resource relations.

Tables 2 and 3 give the exact parameter settings that we use in each of the test sets, which will be explained below. We use a full factorial design: we test each possible combination of parameter settings in each set. If we give a range for a parameter, we randomly draw values from the given range. These values are the same for all settings of the other parameters. For each combination of parameter settings, we generate ten problem instances, in order to obtain a variety of problem instances. Since each parameter setting has a default value (or range) that is the same in each test set, there are ten problem instances that are part of each each set. The parameter settings for these ten problem instances are used below to explain how we generate problem instances. Table 2 gives the values for the parameters that do not vary over a range and Table 3 gives the values for the parameters that do vary. In total, sets 1, 2, and 3 consists of 1,280, 160, and 80 problem instances, respectively. Each problem is solved using a target availability of 95%.

The (symmetrical) repair network consists of a central depot, two intermediate depots, and ten operating sites. Details on the lead times can be found in the tables; for a definition of the lead times, see Section 3.1. The product structure consists of 50 LRUs, 100 SRUs, and 200 parts (more details in Appendix B). The failure rate of a component is equal to the summation of the failure rates of its subcomponents (if any exist) and the failure rate of each LRU is close to the range [0.01; 0.25] (more details in Appendix B). The net component price is in the range [1,000; 10,000] (more details in Appendix B). Using these prices, we calculate the variable costs as follows:

- repair costs are a fraction of the net component price. This fraction is in the range [0.25; 0.75];

Parameter	Default range	Additional range(s)	Varied in test set
Annual demand of LRU	0.01 – 0.25	0.01 – 0.1 & 0.01 – 0.5 & 0.01 – 1	2
Net cost of component	1,000 – 10,000	1,000 – 100,000	2
Discard costs	0.75 – 1.25	—	—
Repair costs	0.25 – 0.75	—	—
Move costs	0.01 – 0.01	—	—
Annual holding costs	0.20 – 0.20	0.20 – 0.40	1
Annual cost of resource	10,000 – 100,000	10,000 – 500,000	2
Discard lead time (in years)	1/10 – 1/2	1/4 – 1/2	1
Repair lead time (in years)	0.5/52 – 4/52	2/52 – 4/52	1
Move lead time (in years)	0.5/52 – 4/52	2/52 – 4/52	1
# Components per resource	2 – 6	2 – 3	3

Table 3: Values that vary over a range in the experiments

Table 4: Fixed values

Parameter	Value(s)
# Echelon levels	2
# Central depots	1
# Operating sites	5
# Indenture levels	1
# LRUs	100
# Resources	10
# Component types	1
% Resources used by 1 component	100%

- we recursively add the costs of each subcomponent to its parent to get the gross component price;
- discard costs are a fraction of the gross component price. This fraction is in the range [0.75; 1.25] and includes the costs for acquiring a new component;
- move costs are 1% of the gross component price;
- annual costs of holding one spare part of a component are 20% of the gross component price.

There are ten resources and their annual costs are in the range [10,000; 100,000] (more details in Appendix B). 50% of the resources are required by one component only, the other 50% are required by 2 to 6 components. We distinguish 4 'component types', for example electronic and mechanical components. Each resource and each LRU family (an LRU including all its subcomponents at any indenture level) is randomly assigned to one of the component types, so that resources of one component type do not interact with resources of another component type, which is realistic in practice.

5.1.2 Two-echelon, single-indenture problem instances

The generator that we described in Section 5.1.1 is an extended version of the generator that Basten et al. (2011a) have used to generate two-echelon, single-indenture problem instances. Therefore, we only give the parameter settings here, in Tables 4 and 5. In total this set consists of $10 \cdot 2^7 = 1{,}280$ problem instances.

Table 5: Values that vary over a range

Parameter	Range(s)
Annual demand of LRU	0.01 – 0.1
Net cost of component	1,000 – 10,000 & 1,000 – 100,000
Discard costs	75% – 125%
Repair costs	25% – 75%
Move costs	1% – 1%
Annual holding costs	20% – 20% & 20% – 40 %
Annual cost of resource	10,000 – 100,000 & 10,000 – 500,000
Discard lead time (in years)	1/10 – 1/2 & 1/4 – 1/2
Repair lead time (in years)	0.5/52 – 4/52 & 2/52 – 4/52
Move lead time (in years)	2/365 – 4/52 & 1/52 – 4/52
# Components per resource	2 – 3 & 2 – 6

Table 6: Overview of the results for the problem instances of Basten et al. (2011a)

Algorithmused	Average achieved availability	Cost reduction compared with sequential average	maximum
Sequential exact	95.14%	–	–
Iterative approximation	95.07%	3.80%	35.46%
Integrated exact	95.11%	5.07%	43.26%
Integrated approximation	95.11%	5.07%	43.26%

5.2 Results

We address the questions that we posed at the start of Section 5. In Section 5.2.1, we compare the results of the iterative algorithm with those of the sequential approach and the integrated algorithm of Basten et al. (2011a) at a high level, and in Section 5.2.2, we analyze how repair strategies change and which parameters influence the results.

5.2.1 Comparison of sequential, iterative, and integrated algorithms

Table 6 gives an overview of the results on the two-echelon, single-indenture test set of Basten et al. (2011a). Compared with solving the two problems sequentially (exact), solving the joint problem using the integrated algorithm (exact) results in a cost reduction of 5.07% on average and more than 43% at maximum. Notice that the results of the exact and approximate version of the integrated algorithm are almost equal (it differs in 2 of the 1,280 problem instances only). We see that the iterative algorithm achieves most of the cost reductions that may be achieved in this test set.

We are mainly interested in the multi-echelon, multi-indenture problem instances. For a fair comparison, we use the approximate evaluation for all three algorithms here. Table 7 gives an overview of the results over the three test sets. A key observation is that the results are in line with the results on the two-echelon, single-indenture problem instances. This suggests that the integrated algorithm finds solutions that are still close to optimal and that the iterative algorithm is very robust. The iterative algorithm achieves a cost reduction of 2.85% on average and almost 35% at maximum compared with the sequential approach. In 9.5% of the problem instances, a cost reduction of over 10% is achieved. The gap with the integrated algorithm is less than 2.5% in over 95% of the problem instances, which means that changes of being a few percent off are very small using the iterative algorithm.

The integrated algorithm requires on average about 35 times as much computation time

Table 7: Overview of the results over the three multi-echelon, multi-indenture test sets

	Computation time in seconds		Achieved availability		Cost reduction compared with			
					sequential		iterative	
	average	maximum	average	maximum	average	maximum	average	maximum
Sequential	0.18	1.83	95.25%	96.32%	—	—	—	—
Iterative	4.50	41.09	95.11%	96.12%	2.85%	34.69%	—	—
Integrated	155.81	10,456.37	95.20%	96.24%	3.40%	36.88%	0.58%	5.26%

Table 8: Cost reduction for important parameter settings

			Average cost reduction		
			iterative	integrated	integrated
Test				compared with	
set	Parameter	Setting	sequential	sequential	iterative
1	# LRUs	50	4.35%	4.98%	0.69%
		100	1.10%	1.46%	0.37%
	Move lead time	[0.5/52; 4/52]	4.59%	5.29%	0.76%
		[2/52; 4/52]	0.86%	1.15%	0.30%
2	Demand per LRU	[0.01; 0.1]	10.96%	12.06%	1.20%
		[0.01; 0.25]	2.47%	2.73%	0.27%
		[0.01; 0.5]	1.32%	2.77%	1.47%
		[0.01; 1]	3.09%	4.61%	1.56%

as the iterative algorithm. At maximum, the integrated algorithm requires almost three hours, which is more than 250 times as much as the iterative algorithm. This clearly shows that the iterative algorithm scales much better (this effect is even stronger for the case study, see Section 6.3).

There are some problem instances for which the integrated algorithm yields higher costs than the iterative approach, at most 2.76%. This is due to the overshoot problem (see Section 2.2 and Section 6.3, especially Figure 5). The integrated algorithm yields a higher availability in these cases as well. There are no problem instances on which the iterative algorithm yields both lower costs and a higher availability than the integrated algorithm. This further strengthens our assumption that the approximate version of the integrated algorithm finds solutions that are close to optimal.

5.2.2 Detailed analysis of repair strategies and important parameters

Here, we focus on the multi-echelon, multi-indenture problem instances only. Table 8 gives the cost reductions for those parameter settings that substantially impact the cost reductions that we achieve. We discuss the results below, including a discussion of the changes in the repair strategies. The other parameters that we varied in test sets 1, 2, and 3 do not have a substantial impact on the achieved cost reductions.

We see that if the difference between the integrated and the sequential algorithm increases, then the difference between the integrated and the iterative algorithm increases as well, but not as fast. So, if the sequential approach leaves more room for improvement, the iterative algorithm will improve more, both absolutely and relatively. Compared with the improvements that the integrated algorithm finds, it seems that the iterative algorithm 'randomly' misses some opportunities for improvement; we are not able to find structural problems in the solutions.

In order to give the best comparison between solving the two problems jointly and sequentially, we only focus on the difference between the integrated and the sequential

Table 9: Detailed results on problem instances consisting of 50 or 100 LRUs

		% of the demand for LRUs that is repaired				% spare LRUs that is located		
	number of LRUs	in total	at echelon level 1	2	3	at echelon level 1	2	3
Sequential	50	94.0%	90.4%	0.0%	3.6%	97.7%	0.7%	1.6%
	100	96.7%	93.4%	0.0%	3.3%	99.0%	0.2%	0.7%
Integrated	50	94.5%	80.7%	4.9%	9.0%	92.8%	3.7%	3.5%
	100	96.9%	89.6%	2.8%	4.5%	97.0%	1.7%	1.3%

Notice: Some problem instances consist of two echelon levels only (echelon levels 1 and 3 in the table).

algorithm below. Table 9 gives some more detailed results on the problem instances consisting of either 50 or 100 LRUs. Before we discuss the differences that result from the difference in number of LRUs, we first discuss how the repair strategies change when solving the joint problem instead of solving the two problems sequentially.

Notice that in the sequential solution, repairs are never performed at the intermediate depots (in test set 1). The reason is that resources are never located at the intermediate depot, probably because they are so expensive that they are interesting only at the central depot. As a result, repairs can be performed at the operating site if no resources are required (why pay more to ship them to the intermediate depot?) or at the central depot if resources are required and available there. Using the integrated algorithm, the numbers change, due to two reasons.

1. More resources are located at the central depot (30% and 16% more for 50 and 100 LRUs, respectively) which means that some components that are discarded in the sequential solution, are now repaired. As a result, the lead time decreases and less spare parts are required.
2. Notice that if repairs of a certain component are performed at the operating sites, then spare components may only be located at the operating sites as well. As a result, some components that do not require any resource in order to be repaired, are repaired at a more central location in the solution of the integrated solution so that risk pooling effects may be used (a spare part can now be located at a more central location and be used at various operating sites).

For the number of LRUs, part of the explanation of the difference in achieved cost reduction is that we do not vary the number of resources in our problem instances, so that there are relatively more LRUs that require a resource in order to be repaired when there are 50 LRUs. This gives more possibilities for cost reductions stemming from option 1 above. Next, we notice that if we increase the target availability for the problem instances with 50 LRUs to 97.5%, the achieved cost reduction reduces to about 2%. 97.5% target availability for 50 LRUs leads to a target availability per LRU that is almost equal to the target availability per LRUs in problem instances consisting of 100 LRUs and having a target availability of 95%. If we then look at the cost reductions that may be achieved for the various values of the demand per LRU, we see that it is lowest for our second lowest setting ([0.01; 0.25]); it is higher when the demand is either lower or higher. It appears that multiple effects (e.g., target availability and demand per LRU) interact, as a result of which there is sometimes a lot to be gained from solving the two problems jointly instead of sequentially, and sometimes not. We are not able to state beforehand which of the two cases will happen.

If the move lead time is relatively low (compared with the repair and discard lead times), changes are high that the sequential approach will lead to a large optimality gap. In that case, the total lead time (repair lead time plus move lead time) when repairing at a higher echelon level is only slightly higher than the repair lead time when repairing at the operating sites. As a result, the disadvantage of repairing at a higher echelon level is relatively small, and the advantage of being able to use risk pooling effects outweighs more often that disadvantage.

For the computation times, all results are as may be expected. Computation times increase using either of the three approaches when the number of indenture levels, number of LRUs, number of echelon levels, or the demand increases. For the integrated algorithm, the computation times also increase when components require more resources on average (maximum number of components per resource increases or the percentage of resources used by one component only decreases).

6 Case study at Thales Nederland

We perform a case study on a sensor system (combined radar and electro-optical surveillance system) manufactured by Thales Nederland. The goal of this study is to find out which cost reduction we may obtain in practice and which advantages and drawbacks of our joint approach we can identify for application in practice. Thales Nederland is part of the Thales Group, which is a high-technology company active in aerospace, space, defense, security, and transportation. Thales Nederland is a manufacturer of naval sensors and naval command and control systems. Since Thales Nederland is active in the defense industry, it is a perfect company for a case study because both the LORA problem and the spare parts stocking problem have been well known problems in the military world for decades. Thales' customers include many navies, e.g., the Royal Netherlands Navy. If such a navy acquires a set of sensor systems, it also demands a plan on how to maintain the systems, which includes a LORA and a recommended spares list. In Section 6.1, we discuss how logistic engineers at Thales Nederland solve the LORA and spare parts stocking problems, and the difficulties this leads to. We give the technical details of the case study in Section 6.2, and in Section 6.3, we compare the results of the iterative algorithms with those of the sequential and integrated algorithms, and those of the logistic engineers.

6.1 Current practice

The logistic engineers at Thales Nederland first conduct a so-called *non-economic LORA*. The goal of this non-economic LORA is to exclude non-realistic repair or discard options and to simplify the problem. Questions that are posed are, for example:

- Is the component prone to failure? Casings, for example, are not under normal circumstances and are therefore not considered in the LORA.
- Does the customer prescribe the maintenance policy for the component? If so, this policy is followed.
- Does the value of the component exceed a certain threshold? If not, it can be discarded by default, since it is not worth repairing.

The result is that for some components, repair/discard options are excluded. If one option only remains, no decision needs to be taken in the LORA problem for that component, but the component is still taken into account in the spare parts stocking problem, because of its influence on the availability. Furthermore, only part of the resources are included in the LORA, mainly the expensive ones.

After the input data has been structured and filtered, the logistic engineer finds a first solution, using decisions made for previous products, his experience, and spreadsheet calculations. Then, he uses a spare parts stocking tool (INVENTRI, based on VARI-METRIC and the work of Rustenburg 2000) to stock spare parts. Analyzing the results, he finds components for which spare parts holding costs are very high. If he thinks that it might help to change the LORA decision for these components, he does so and calculates the new LORA costs and solves a spare parts stocking problem. So in fact, the logistic engineer tries to perform some manual iterations, which has as its drawbacks that this approach is:

- time consuming, since an analysis takes up to a few days after all data has been acquired;
- unreproducible, because of the judgmental feedback loop;

- error sensitive.

Usage of the iterative (or integrated) algorithm would take away these drawbacks.

6.2 Case: a sensor system

Although the considered product structure consists of six indenture levels, we consider only three indenture levels, as a result of the non-economic LORA. For the same reason, the product structure consists of over 1,500 components, but only slightly more than 200 turn out to be relevant, of which 40% are LRUs. For about one third of the components, only one repair/discard option remains, and for an additional one third, the repair/discard options that can be chosen are restricted. The repair network consists of twelve ships, attached to two intermediate depots, a central depot and the OEM (spare parts may not be stocked at the OEM and if repairs are performed at the OEM, then the variable repair costs per repair action are higher, but an investment in resources is not required for the navy). There are 54 resources, 34 of which are 'adapters'. These adapters are used in concurrence with other test equipment.

The costs of the various components can be up to one million euros, and the costs of the various resources can be up to a couple of million euros. These costs are not used directly. Instead, there are three types of costs in the joint problem of LORA and spare parts stocking: variable costs per repair or discard action, fixed annual costs for locating resources, and annual spare parts holding costs. For each type of costs, we give the most important cost factors that we include (another overview of the cost factors to include can be found in Saranga and Dinesh Kumar 2006):

- Variable repair costs (customer's network): working hours (e.g., locating failure, exchanging subcomponents, and performing direct repair), variable costs for using resources (e.g., energy consumption and wear), and usage of additional parts (e.g., bulk items such as screws and wires).
- Variable repair costs (OEM and outsourced in general): listed repair price.
- Variable discard costs: procurement price for the component that replaces the discarded component and disposal costs or a residual value of the discarded component.
- Variable move costs: transportation, handling, and administrative costs.
- Fixed resource costs: depreciation costs, costs of capital, a risk factor (e.g., insurance against damage and theft), fixed operating costs (e.g., a location to operate the equipment), and maintenance costs of the resource. Resources may have a residual value after their economic lifetime.
- Spare parts holding costs: costs of capital, a risk factor, and storage costs. Spares may have a residual value after the lifetime of the product.

6.3 Results

Figure 4 shows the results on the case study, with a target availability of 95%. We see that the iterative algorithm finds the best solution with a cost reduction of 9.7%. The reason that the solution of the integrated algorithm leads to higher costs is the overshoot problem, as can be seen in Figure 5. We also see that some parts of the curve are quite dense, whereas others are not. This means that if we would have aimed for an availability of, for example, 94.8%, the overshoot would be very low. The figure also shows that the iterative algorithm finds a solution that is very close to the curve of solutions that the integrated algorithm finds, and therefore probably very close to optimal.

The iterative algorithm requires less than one minute (11 iterations), whereas the integrated algorithm requires almost two days. This means that for usage at Thales Nederland, the iterative algorithm fits best.

The cost reductions are achieved as follows. More resources are installed and more repairs are performed in the customer's network and in total. This leads to higher resource costs and

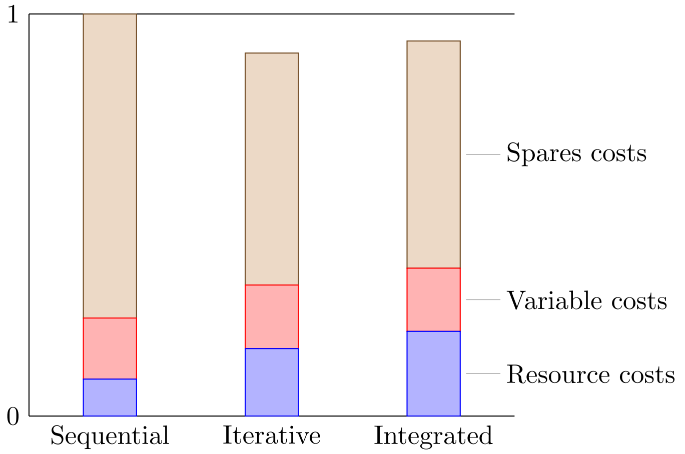

Figure 4: Costs for Thales case (normalised)

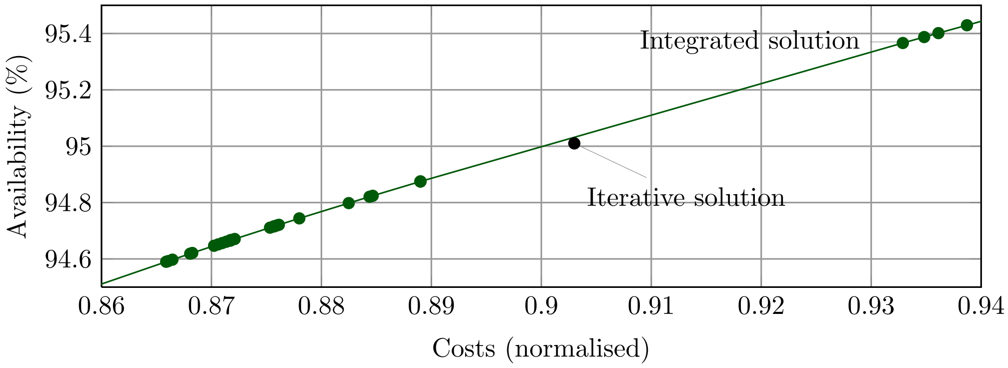

Figure 5: Availability as a function of total costs for the case study

higher variable costs, but also to much lower spare parts holding costs. The cost reduction is achieved by:

- installing two resources at the depot that are not installed in the sequential solution;
- installing one resource at both intermediate depots instead of one at the central depot;
- installing one resource at all ships instead of one at each of the two intermediate depots.

The other resources are installed at the same echelon level in both solutions. The logistic engineer at Thales Nederland achieves about a quarter of the cost reduction that we achieve using the iterative algorithm. He locates resources at a more central location than our iterative algorithm does, but at a more decentralized location than the sequential algorithm does.

7 Conclusions and further research

In this paper, we presented an iterative algorithm for the joint problem of LORA and spare parts stocking for multi-indenture, multi-echelon problem instances with very mild restrictions on the resource-component relations. Our conclusions are as follows:

- The iterative algorithm performs very good on average. Only in rare cases do cost differences of a few percent exist with the integrated method.
- The difference between the approximate and exact version of the integrated algorithm is very small.

- The iterative algorithm scales very well. Computation time is not a problem, whereas it is a huge problem for the integrated algorithm
- The iterative algorithm can be used in practice and leads to a substantial cost reduction compared to solving the two problems sequentially. In fact, the algorithm has been adopted by Thales Nederland.
- The iterative algorithm may easily be extended if extensions affect only one building block. An example would be certain flexibility options in the spare parts stocking analysis, e.g., lateral transshipments or emergency shipments.

Our recommendations for further research are as follows:

- Extend the model so that the exact repair network can be modeled. In both the LORA and the spare parts stocking problem this is easily done, so the key problem to solve is how to decompose the spare parts holding costs so that they can be fed back to the LORA.
- Include finite repair capacities in the model. This is already difficult for the spare parts stocking problem alone, but there is some literature available (see, e.g., Sleptchenko et al. 2002). The feedback mechanism changes since the spare parts holding costs are now not attached to one possible repair/discard decision, but to a possible repair/-discard decision including a number of resources (in case of repair). However, we do not expect too much problems with this change in the feedback mechanism.

Many practically relevant extensions such as a probability of unsuccessful repair and the aforementioned flexibility options in the spare parts stocking analysis may be included as well. Since they do not affect the feedback mechanism, they will not cause problems and are therefore interesting from a business point of view, but probably not from an academical point of view.

With all possible extensions, it will be hard to compare the iterative algorithm with the integrated algorithm, since the computation time of the latter algorithm will explode.

Acknowledgments

The authors gratefully acknowledge the support of the Innovation-Oriented Research Programme 'Integral Product Creation and Realization (IOP IPCR)' of the Netherlands Ministry of Economic Affairs, Agriculture and Innovation.

A Termination of the iterative algorithm

We proof that the iterative algorithm necessarily converges in a finite number of iterations. We do so by showing that the number of possible solutions is finite and that it is not possible to have a cycle (a number of iterations after which we find exactly the same solution). In our proof we assume that the discard option is available at each echelon level to decrease the notational complexity. The proof is straightforwardly adapted if discard is available at the central depot only.

Lemma 1. *The number of possible solutions is finite.*

Proof. If there are $|E|$ echelon levels, then each resource can be located at the various echelon levels in $2^{|E|}$ ways. Since there are $|R|$ resources, the total number of combinations for the resource locations is $2^{|E|^{|R|}}$.

Each component may be repaired or discarded at any of the $|E|$ echelon levels, so there are $2 \cdot |E|$ repair/discard options per component. Since there are $|C|$ components, there are $(2 \cdot |E|)^{|C|}$ options in total. In fact, there are generally less combinations, since some repair/discard options are unavailable for a subcomponent given the repair/discard decision for its parent component.

Combining the options for the resources and the components shows that there are at most $2^{|E|^{|R|}} \cdot (2 \cdot |E|)^{|C|}$ possible solutions for the LORA problem. Due to the exact allocation of resources to echelon levels, some repair options may not be available for certain components.

Since a given solution for the LORA problem leads to one specific allocation of spare parts in the spare parts stocking problem, there are also at most $2^{|E|^{|R|}} \cdot (2 \cdot |E|)^{|C|}$ solutions for the joint problem. □

We define $\mathcal{H}^i$ as the three-dimensional array with entries $hc^i_{c,e,d}$ in iteration i.

Corollary 1. *$\mathcal{H}^i$ can be filled in at most $(|C| \cdot |E| \cdot 2)^{1+2^{|E|^{|R|}} \cdot (2 \cdot |E|)^{|C|}}$ ways.*

Proof. As a result of Lemma 1, each entry $hc^i_{c,e,d}$ in $\mathcal{H}^i$ can be filled in at most $2^{|E|^{|R|}} \cdot (2 \cdot |E|)^{|C|}$ ways in each iteration i ($i > 1$). In addition, each $hc^1_{c,e,d} = 0$. Since there are $|C| \cdot |E| \cdot 2$ values to be filled, $\mathcal{H}^i$ can be filled in at most $(|C| \cdot |E| \cdot 2)^{1+2^{|E|^{|R|}} \cdot (2 \cdot |E|)^{|C|}}$ ways. □

We now show that cycling cannot occur, which is formalized in Lemma 2. In iteration i, an estimate of the spare parts holding costs is included in the inputs for the LORA problem ($\mathcal{H}^i$). The iteration starts with solving the LORA problem, leading to a certain LORA solution LS^i (the repair/discard decisions and resource locations) with resulting LORA costs LC^i (variable and fixed costs) and resulting estimated spare parts holding costs EC^i. Next, the spare parts stocking problem is solved. The total costs of the solution in iteration i are now the LORA costs (excluding the estimated spare parts holding costs) LC^i plus the spare parts holding costs resulting from the spare parts stocking problem SC^i. In iteration $i+1$, the estimates of the spare parts holding costs in the LORA inputs are adapted again (to $\mathcal{H}^{i+1}$) and a new LORA solution is found LS^{i+1}, with associated costs $LC^{i+1} + EC^{i+1}$. The proof of Lemma 2 states that cycling can occur only if $LC^{i+1} + EC^{i+1}$ is equal to $LC^i + SC^i$. In other words, the solution that is found in iteration $i+1$ may be different from the solution that is found in iteration i, but they lead to the same costs using the LORA inputs in iteration $i+1$ (including $\mathcal{H}^{i+1}$). Therefore, we could just as well pick solution LS^i again and terminate the algorithm, which is what we describe in Section 4.

Lemma 2. *The iterative algorithm cannot cycle between a number of realizations of $\mathcal{H}^i$.*

Proof. We define $X^i_{c,e,d}$, $Y^i_{r,e}$, and $S^i_{c,e}$ as the realization in iteration i of $X_{c,e,d}$, $Y_{r,e}$, and $S_{c,e}$, respectively. Then, we define $LC^i = \sum_{c \in C} \sum_{e \in E} \sum_{d \in D_e} vc_{c,e,d} \cdot \lambda_c \cdot X^i_{c,e,d} + \sum_{r \in R} \sum_{e \in E} fc_{r,e} \cdot Y^i_{r,e}$, $EC^i = \sum_{c \in C} \sum_{e \in E} \sum_{d \in \{\text{discard,repair}\}} hc^i_{c,e,d} \cdot \lambda_c \cdot X^i_{c,e,d}$, and $SC^i = \sum_{c \in C} \sum_{f \in E} hc'_{c,f} \cdot S^i_{c,f}$. EC^i represents the estimated spare parts holding costs in the LORA solution in iteration i, whereas SC^i represents the actual spare parts holding costs resulting from solving the spare parts stocking problem in iteration i. Remember from Section 4 that for each tuple (c, e, d) (with $d \in \{\text{repair, discard}\}$) for which $X^{i-1}_{c,e,d} > 0$, we set $hc^i_{c,e,d} = \frac{\sum_{f \in E} hc'_{c,f} \cdot S^{i-1}_{c,f}}{\lambda_c}$. Therefore $SC^i = \sum_{c \in C} \sum_{f \in E} hc'_{c,f} \cdot S^i_{c,f} = \sum_{c \in C} \sum_{e \in E} \sum_{d \in \{\text{discard,repair}\}} \lambda_c \cdot hc^{i+1}_{c,e,d} \cdot X^i_{c,e,d}$. Finally, we define Δ^i as the difference between SC^i and EC^i, so $\Delta^i = SC^i - EC^i = \sum_{c \in C} \sum_{e \in E} \sum_{d \in \{\text{discard,repair}\}} \left(hc^{i+1}_{c,e,d} - hc^i_{c,e,d} \right) \cdot \lambda_c \cdot X^i_{c,e,d}$.

If $LC^{i+1} + EC^{i+1} = LC^i + SC^i$, we know that $LS^{i+1} = LS^i$ and the algorithm terminates. So, in a cycle of length n ($n > 1$) it should hold that $LC^{i+1} + EC^{i+1} < LC^i + SC^i = LC^i + EC^i + \Delta^i$. For iteration $i+2$ it then holds that $LC^{i+2} + EC^{i+2} < LC^{i+1} + EC^{i+1} + \Delta^{i+1}$, so $LC^{i+2} + EC^{i+2} < LC^{i+1} + EC^{i+1} + \Delta^{i+1} < LC^i + EC^i + \Delta^i + \Delta^{i+1}$. For the n^{th} iteration, we get $LC^{i+n} + EC^{i+n} < LC^i + EC^i + \sum_{k=0}^{n-1} \Delta^{i+k}$. In a cycle of length n ($n > 1$), we also know that $LC^{i+n} = LC^i$ and $EC^{i+n} = EC^i$, so it should hold that $\sum_{k=0}^{n-1} \Delta^{i+k} > 0$. However, $\sum_{k=0}^{n-1} \Delta^{i+k} = \sum_{k=0}^{n-1} \sum_{c \in C} \sum_{e \in E} \sum_{d \in \{\text{discard,repair}\}} \left(hc^{i+k+1}_{c,e,d} - hc^{i+k}_{c,e,d} \right) \cdot \lambda_c \cdot X^{i+k}_{c,e,d} = \sum_{c \in C} \sum_{e \in E} \sum_{d \in \{\text{discard,repair}\}} \sum_{k=0}^{n-1} \left(hc^{i+k+1}_{c,e,d} - hc^{i+k}_{c,e,d} \right) \cdot \lambda_c \cdot X^{i+k}_{c,e,d}$. We know that if $X_{c,e,d}^{i+k} = 0$, then $hc_{c,e,d}^{i+k+1} = hc_{c,e,d}^{i+k}$. We also know that $hc_{c,e,d}^{i+n} = hc_{c,e,d}^{i}$. It is now easily seen that $\sum_{k=0}^{n-1} \left(hc_{c,e,d}^{i+k+1} - hc_{c,e,d}^{i+k} \right) \cdot \lambda_c \cdot X_{c,e,d}^{i+k} = 0$ for all $c \in C$, $e \in E$, $d \in \{\text{discard}, \text{repair}\}$ and therefore $\sum_{k=0}^{n-1} \Delta^{i+k} = 0$. We have shown that a cycle cannot occur if we only pick another LORA solution in iteration $i + 1$ if that leads to a strictly better result than using the LORA solution of iteration i again. Therefore, the iterative algorithm cannot cycle. □

Theorem 1. *The iterative algorithm always terminates after a finite number of iterations.*

Proof. This holds since the number of possible matrices $\mathcal{H}^i$ is finite, as stated in Corollary 1, and the algorithm cannot cycle between a number of realizations of $\mathcal{H}^i$, as stated in Lemma 2. □

B Problem instances generator

In this appendix, we supply additional details on the generation of the product structure, failure rates, component prices, and resources prices.

The three-indenture product structure consists of 50 LRUs, 100 SRUs, and 200 parts (so two subcomponents per component on average). Each subcomponent is randomly assigned to one of the components at the next lower indenture level. As a result, in general, the number of subcomponents per component differs for the various components. The annual demand for a component is equal to the annual demand of its subcomponents (if any exist). We achieve this by drawing the annual demand of each part from a uniform distribution on the interval [0.01/(#subcomp. per comp.)^2; 0.25/(#subcomp. per comp.)^2], and recursively calculating the annual demand of the SRUs and LRUs. The demand for SRUs or LRUs without subcomponents is drawn from the same interval as the demand for parts.

For each component, we draw a net price from a shifted exponential distribution with shift factor 1,000 and rate parameter 7/(10,000 − 1,000). As a result, we do not have components with a price below 1,000, since they are typically discarded by default. Furthermore, there are considerably more cheap components than expensive ones. On average 1‰ of the components get a value larger than 10,000, but we draw a new price for these components to avoid odd problem instances. The annual costs of the resources are drawn from a shifted exponential distribution with shift factor 10,000 and rate parameter 7/(100,000 − 10,000). To avoid odd problem instances, we draw a new value if we have drawn a value higher than 100,000.

References

Alfredsson, P. (1997). Optimization of multi-echelon repairable item inventory systems with simultaneous location of repair facilities. *European Journal of Operational Research*, 99:584–595.

Barros, L. L. (1998). The optimization of repair decisions using life-cycle cost parameters. *IMA Journal of Mathematics Applied in Business & Industry*, 9:403–413.

Barros, L. L. and Riley, M. (2001). A combinatorial approach to level of repair analysis. *European Journal of Operational Research*, 129(2):242–251.

Basten, R. (2010). *Designing logistics support systems. Level of repair analysis and spare parts inventories.* PhD thesis, University of Twente, Enschede (The Netherlands).

Basten, R. J. I., Schutten, J. M. J., and van der Heijden, M. C. (2009). An efficient model formulation for level of repair analysis. *Annals of Operations Research*, 172(1):119–142.

Basten, R. J. I., Van der Heijden, M. C., and Schutten, J. M. J. (2010). A minimum cost flow model for level of repair analysis. *International Journal of Production Economics.* To appear. DOI: 10.1016/j.ijpe.2010.03.025.

Basten, R. J. I., Van der Heijden, M. C., and Schutten, J. M. J. (2011a). Joint optimization of level of repair analysis and spare parts stocks. BETA working paper.

Basten, R. J. I., Van der Heijden, M. C., and Schutten, J. M. J. (2011b). Practical extensions to a minimum cost flow model for level of repair analysis. *European Journal of Operational Research*, 211:333–342.

Brick, E. S. and Uchoa, E. (2009). A facility location and installation or resources model for level of repair analysis. *European Journal of Operational Research*, 192(2):479–486.

Graves, S. C. (1985). A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management Science*, 31(10):1247–1256.

Muckstadt, J. A. (1973). A model for a multi-item, multi-echelon, multi-indenture inventory system. *Management Science*, 20(4):472–481.

Muckstadt, J. A. (2005). *Analysis and Algorithms for Service Parts Supply Chains*. Springer, New York (NY).

Rustenburg, W. D. (2000). *A System Approach to Budget-Constrained Spare Parts*. PhD thesis, Eindhoven University of Technology, Eindhoven (The Netherlands).

Rustenburg, W. D., van Houtum, G. J., and Zijm, W. H. M. (2003). Exact and approximate analysis of multi-echelon, multi-indenture spare parts systems with commonality. In Shanthikumar, J. G., Yao, D. D., and Zijm, W. H. M., editors, *Stochastic Modelling and Optimization of Manufacturing Systems and Supply Chains*, pages 143–176. Kluwer, Boston (MA).

Saranga, H. and Dinesh Kumar, U. (2006). Optimization of aircraft maintenance/support infrastructure using genetic algorithms — level of repair analysis. *Annals of Operations Research*, 143(1):91–106.

Sherbrooke, C. C. (1968). METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, 16(1):122–141.

Sherbrooke, C. C. (1986). VARI-METRIC: Improved approximations for multi-indenture, multi-echelon availability models. *Operations Research*, 34:311–319.

Sherbrooke, C. C. (2004). *Optimal inventory modelling of systems. Multi-echelon techniques.* Kluwer, Dordrecht (The Netherlands), second edition.

Sleptchenko, A., Van der Heijden, M. C., and Van Harten, A. (2002). Effects of finite repair capacity in multi-echelon, multi-indenture service part supply systems. *International Journal of Production Economics*, 79:209–230.

Working Papers Beta 2009 - 2011

nr.	Year	Title	Author(s)
347	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	Joint optimization of level of repair analysis and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Inventory control with manufacturing lead time flexibility	Ton G. de Kok
344	2011	Analysis of resource pooling games via a new extenstion of the Erlang loss function	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Vehicle refueling with limited resources	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger
342	2011	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Bilge Atasoy, Refik Güllü, TarkanTan
341	2011	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Analysis of a two-echelon inventory system with two supply modes	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
334	2010	Flexible Heuristics Miner (FHM)	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan

330	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	Preventing or escaping the suppression mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
319	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Minimal Protocol Adaptors for Interacting Services	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	Teaching Retail Operations in Business and Engineering Schools	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.

		Design for Availability: Creating Value for Manufacturers and Customers	Pieter van Gorp, Rik Eshuis.
316	2010		
		Transforming Process Models: executable rewrite rules versus a formalized Java program	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
315	2010		
		Getting trapped in the suppression of exploration: A simulation model	S. Dabia, T. van Woensel, A.G. de Kok
314	2010		
313	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
	2010		
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten

297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok

277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	R. Seguel, R. Eshuis, P. Grefen
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl