# Practical extensions to the level of repair analysis

February 17, 2010

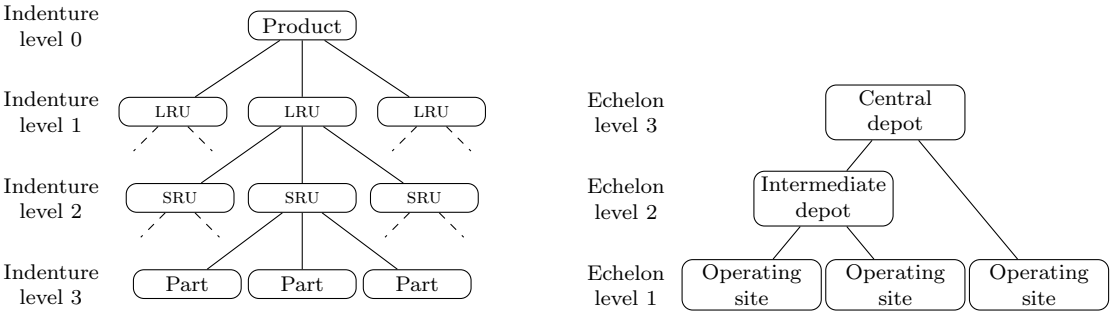
**Abstract**

The level of repair analysis (LORA) gives answers to three questions that are posed when deciding on how to maintain capital goods: 1) which components to repair upon failure and which to discard, 2) at which locations in the repair network to perform each type of repairs, and 3) at which locations in the network to deploy resources, such as test equipment. The goal is to achieve the lowest possible life cycle costs. Various models exist for the LORA problem. However, these models tend to be restrictive in that specific business situations cannot be incorporated, for example, having repair equipment with a capacity restriction or the occurrence of unsuccessful repairs. We discuss and model various practically relevant extensions to an existing minimum cost flow formulation for the LORA problem. We show the added value of these model refinements in an extensive numerical experiment.

## 1 Introduction

Most probably, failures will occur during the life time of manufactured products and installations. Inexpensive products, such as many consumer goods, will be discarded upon failure. Capital goods, which are more expensive, will be repaired. Capital goods are physical systems that are used to produce products or services and have high downtime costs. Examples are manufacturing systems, defence systems, medical devices, and airplanes.

Since their downtime costs are high, capital goods are typically maintained using a *repair by replacement* policy in which the failed component is taken out of the system and replaced by a functioning spare part. The defective component, called an *LRU* or *line replaceable unit* in the military world, can either be discarded or repaired. In the former case, a new component needs to be acquired. In the latter case, the component may be repaired directly or by replacement of a subcomponent. Since such a replacement is typically performed in a repair shop, these subcomponents at the *second indenture*




(a) Product structure

(b) Repair network

Figure 1: Examples, including the naming convention that we use

*level* are called *SRUs* or *shop replaceable units*. Figure 1a gives an example of a multi-indenture product structure including the naming convention that we use. If a component is repaired, it should also be decided where to do that in the *multi-echelon repair network*. An example is shown in Figure 1b, including the naming convention that we use.

The level of repair analysis (LORA) gives answers to three questions, the first two of which are called the *repair/discard decisions*:

1. which components to repair upon failure and which to discard,
2. at which locations in the network to perform each type of repairs and discards, and
3. at which locations to deploy resources, such as test equipment,

such that the lowest costs over the complete life cycle of the product are achieved. Those costs consist of both fixed costs and costs that are variable in the number of failures. Variable costs include costs of hiring service engineers, transportation of components, and usage of bulk items; fixed costs include costs for resources such as repair equipment.

The number and locations of spare parts that need to be stocked in the network to achieve a certain availability of the installed base are not considered in the LORA, but in a spare parts stocking problem that is solved after the LORA has been solved. In the context of capital goods, typically METRIC type methods are used to solve this problem (see, e.g., Sherbrooke, 2004; Muckstadt, 2005).

The outline of this paper is as follows. In Section 2, we present related literature and show the contribution of our paper. In Section 3, we describe the LORA problem. Next, in Section 4, we give the minimum cost flow formulation that we base our extensions on. In Section 5, we motivate our choice for the various extensions that we model in Section 6. We perform an extensive computational experiment in Section 7 and in Section 8, we draw conclusions and give recommendations for further research.

## 2 Literature and contribution

In each of the papers that focus purely on the LORA, a (mixed) integer linear programming model is used. Barros (1998) and Barros and Riley (2001) use a model in which two-indenture, two-echelon problem instances are assumed. However, the model can easily be extended to general multi-echelon and multi-indenture problem instances. All components at one indenture level require the same uncapacitated resource in order to be repaired. Furthermore, all data is aggregated per echelon level, which means that the same decision is taken for each location at one echelon level.

Saranga and Dinesh Kumar (2006) assume three-indenture product structures and three-echelon repair networks, but the extension to general multi-indenture, multi-echelon problem instances is straightforward. Resources are not shared between multiple components, and each component requires exactly one unique uncapacitated resource. Again, all data is aggregated per echelon level.

Basten et al. (2009b) generalize the aforementioned models in that fairly general resource-component relations can be modelled: multiple components may share resources and one component may require multiple resources simultaneously. Multi-indenture, multi-echelon problem instances can be modelled, but data is still aggregated per echelon level.

Basten et al. (2008) model the exact repair network and reformulate the LORA problem as a minimum cost flow model with side constraints. This model appears to be very efficient. For example, compared to the model of Basten et al. (2009b), the computation time decreases with over 75% for the largest problem instances in the computational experiments. The authors also claim that their model is flexible in that practically relevant extensions can easily be modelled. We come back to this below.

Besides the papers that focus purely on the LORA, also four papers exist in which the LORA is combined with another problem. Since combining two problems increases the complexity, simplifying assumptions are made in these papers. Brick and Uchoa (2009) add to the LORA problem the problem of where to locate repair facilities, the facility location problem (see, e.g., Daskin, 1995). The authors assume single-echelon repair networks and two-indenture product structures.

Alfredsson (1997) presents a model and solution technique for the joint problem of LORA and spare parts stocking. The author assumes one indenture level only and two echelon levels, but the extension to more echelon levels is straightforward. Basten et al. (2009c) present another model for the same problem. The authors assume general multi-indenture, multi-echelon problem instances and present an iterative solution technique to solve the model: they solve a LORA problem, then a spare parts stocking problem, adapt the LORA inputs based on the results of the spare parts stocking problem, solve the LORA

problem again, et cetera. Basten et al. (2009a) use the same model and present a different solution technique, based on the solution technique of Alfredsson (1997).

An improved LORA model, which we develop, can be used in an iterative method to solve the joint problem of LORA and spare parts stocking, such as presented by Basten et al. (2009c). The contribution of our paper is two-fold.

1. We model practically relevant extensions to the LORA model, thus showing that the minimum cost flow formulation of Basten et al. (2008) is very flexible indeed. We discuss a probability of unsuccessful repairs, capacitated resources, multiple failure modes per component, a probability that no failure is detected in a component that is sent to be repaired (a no-fault-found), and the option to outsource repairs. Based on our contacts with industry representatives and our experience at Thales Nederland, a manufacturer of naval sensors and naval command and control systems, we believe that these are the most important extensions that are required in practice.
2. Using a numerical experiment, we test the impact of inclusion of the model refinements on the repair strategies, the resulting costs, and the computation time. If repair strategies (LORA decisions) change, the model refinements should be incorporated, even though this requires more input data and more computation time. If only the costs change, it may be possible not to include the model refinements, find a good repair strategy using a simple model, and adapt the costs afterwards.

# 3 Problem description and notation

We model a divergent multi-echelon repair network. Therefore, each location in the network has a single upstream location, see, e.g., Figure 1b. We further model a multi-indenture product structure, see, e.g., Figure 1a. We assume that no commonality exists, so one subcomponent cannot be part of two different components.

The system itself (indenture level 0) is never moved from its location, but is repaired by replacement of an LRU. In general, there are three possible decisions $d \in D$ to take for a failed component at a certain location:

- Discard: the component is scrapped and a new one is acquired.
- Repair: the component is repaired by replacing one or more defective subcomponents with operating ones, or by repairing the component directly. A decision needs to be taken for each subcomponent (if any) at the same location.
- Move: the component is moved to the parent location, where a new decision needs to be taken. Obviously, this option is not available at the most central location.

$L$ is the set of locations and $L_1$ is the set of operating sites (echelon level 1). $\Phi_l$ is the set of locations that supply failed components to location $l$. Let the set $C$ consist of all

components and let $C_1$ be the set of LRUs (indenture level 1).

$\lambda_{c,l}$ is defined only for LRUs at operating sites: it is the average annual number of failures in LRU $c \in C_1$ at operating site $l \in L_1$. $\Gamma_c$ denotes the set of subcomponents of component $c$ at the next higher indenture level. We define $q_{c,b}$ as the fraction of failures in component $c$ that is due to a failure in component $b$ ($b \in \Gamma_c$).

Resources $r \in R$ may be required to enable discard, repair, or move. $\Omega_r$ consists of all combinations of a component $c$ and an action $d$ for which resource $r$ is required. So, if component $c$ requires resource $r$ in order to enable action $d$, then the 2-tuple $(c, d) \in \Omega_r$

$vc_{c,l,d}$ are the variable costs of taking action $d$ for one component $c$ at location $l$. Without loss of generality, we have chosen to minimize the average total costs per year with our definition of $\lambda_{c,l}$. Therefore, we define $fc_{r,l}$ to be the annual fixed costs to locate resource $r$ at location $l$.

# 4 Minimum cost flow model

In this section, we explain, based on Basten et al. (2008), how the LORA problem can be modelled as a minimum cost flow model with side constraints (see, e.g., Ahuja et al., 1993, for an overview of minimum cost flow models).

To define the graph $G = (V, A)$ underlying the flow model, with $V$ being the set of nodes in the graph and $A$ being the set of arcs in the graph, we need four different node types (source, decision, transformation, and sink nodes), which we discuss in Section 4.1. In Section 4.2, we discuss how to model the use of resources. Finally, we provide the formal model formulation in Section 4.3.

## 4.1 Nodes

The flow out of a *source node* represents occurrences of failures of an LRU at an operating site. So, for every LRU $c \in C_1$ and every operating site $l \in L_1$, there is one source node $v \in V^{\mathrm{s}} \subseteq V$ with outflow $o_v := \lambda_{c,l}$.

An arc originating at a source node terminates at a *decision node* $v \in V^{\mathrm{d}} \subseteq V$. An example is provided in Figure 2. Every arc going out of a decision node represents one of the possible decisions that can be taken: discard, repair, or move. The variable costs $ac_{v,w}$ for using an arc $(v, w)$ with $v \in V^{\mathrm{d}}$ are equal to $vc_{c,l,d}$, where arc $(v, w)$ represents decision $d$ for component $c$ at location $l$. All arcs that do not originate at a decision node have zero costs associated to them ($ac_{v,w} = 0$ for all $(v, w) \in A$ with $v \notin V^{\mathrm{d}}$).

The arc representing moving component $c$ at location $k$ terminates at the decision node representing component $c$ at location $l$ if $k \in \Phi_l$. The arcs representing repair and

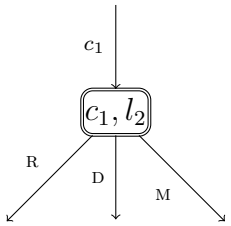



Figure 2: Decision node
A decision needs to be taken for failures of component $c_1$ at location $l_2$. *Node*: A tuple (component, location). *Arcs*: A component for which no decision has been taken yet, or a letter representing a decision that can be taken: R represents *repair*, D represents *discard*, and M represents *move*.

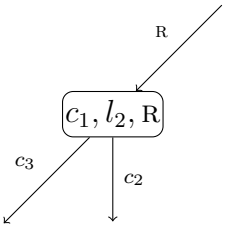



Figure 3: Transformation node
Component $c_1$ is repaired and $\Gamma_{c_1} = \{c_2, c_3\}$. *Node*: A tuple (component, location, state), with state R representing *repair is chosen*. *Arcs*: A component for which no decision has been taken yet, or a letter representing a state.

discard terminate at a transformation node and sink node, respectively.

*Transformation nodes* $v \in V^{\mathrm{t}} \subseteq V$ represent the repair of a component (see Figure 3 for an example). If arc $(v, w)$ represents component $b \in \Gamma_c$ resulting from repairing component $c$, then the fraction of inflow in $v$ that flows out on arc $(v, w)$ is $p_{v,w} := q_{c,b}$.

If no other decisions need to be taken after a certain decision has been made (e.g., discard has been chosen), then the flow goes to a *sink node*.

## 4.2 Modelling resources

In some cases, resources $r \in R$ are required before a certain decision can be taken for certain components. $\Theta_{r,l}$ is the set of arcs that are enabled due to the location of resource $r$ at location $l$. So, for each location $l$ and each resource $r$ we define $\Theta_{r,l} = \{(v, w) \mid (v, w)$ denotes decision $d$ for component $c$ at location $l$ and $(c, d) \in \Omega_r\}$. Notice that fixed costs are thus related to the arcs originating at the decision nodes only.

## 4.3 Flow model formulation

There are two sets of decision variables:

$$F_{v,w} = \text{the amount of flow through arc } (v, w)$$

$$Y_{r,l} = \begin{cases} 1, & \text{if resource } r \text{ is located at location } l \\ 0, & \text{otherwise} \end{cases}$$

Furthermore, we introduce a big $M$ that is used to make those resources uncapacitated for which all required resources (if any) are available. The value of the big $M$ is set for each arc $(v, w)$ such that it is equal to the maximum possible value of $F_{v,w}$. The resulting minimum cost flow model with side constraints is:

$$\text{minimize} \sum_{(v,w)\in A} ac_{v,w} \cdot F_{v,w} + \sum_{r\in R}\sum_{l\in L} fc_{r,l} \cdot Y_{r,l} \tag{1}$$

subject to:

$$F_{v,w} = o_v, \forall (v,w) \in A \mid v \in V^{\mathrm{s}} \tag{2}$$

$$\sum_{u|(u,v)\in A} F_{u,v} = \sum_{w|(v,w)\in A} F_{v,w}, \forall v \in V^{\mathrm{d}} \tag{3}$$

$$F_{v,w} = p_{v,w} \cdot \sum_{u|(u,v)\in A} F_{u,v}, \forall (v,w) \in A \mid v \in V^{\mathrm{t}} \tag{4}$$

$$F_{v,w} \leq M \cdot Y_{r,l}, \forall r \in R, \forall l \in L, \forall (v,w) \in \Theta_{r,l} \tag{5}$$

$$F_{v,w} \geq 0, \forall (v,w) \in A \tag{6}$$

$$Y_{r,l} \in \{0,1\}, \forall r \in R, \forall l \in L \tag{7}$$

Constraint 2 states that the outflow of each source node $v$ is equal to $o_v$ and Constraint 3 assures that the inflow into any decision node is equal to the outflow (balancing constraint). For any arc $(v,w)$ going out of a transformation node, Constraint 4 assures that the inflow into that node is transformed to outflow on arcs $(v,w)$. Constraint 5 assures that only arcs are used that are enabled due to the availability of resources.

# 5 Motivation of model extensions

In this section, we motivate the extensions that we model in Section 6. In Sections 5.1 to 5.5, we discuss the possibility of unsuccessful repairs, the possibility of a no-fault-found, capacitated resources, the possibility of having multiple failure modes in one type of component, and the option to outsource repairs, respectively.

## 5.1 Unsuccessful repair

In the basic LORA model, it is assumed that all repairs are successful. In practice, however, repairs may be unsuccessful. First, a failure can occassionally be very serious, yielding unrecoverable damage. In that case, the component has to be discarded. Second, components can generally not be repaired over and over again. After they have been repaired a number of times, they can only be discarded. Third, it is possible that a failure is occasionally rather complex, so that specific expertise is required. In that case, repair at the operating site may be unsuccessful, but a second repair attempt at the central depot may be successful. At Thales Nederland, a rule of thumb is that about 5% of the repairs

is unsuccessful and results in discarding the component.

A component that needs to be discarded after a failed repair attempt, is not necessarily discarded at the location where the repair attempt was made. If the discard costs differ at the various echelon levels, it may be cost-effective to ship and discard the component upstream. We expect that inclusion of the probability of unsuccessful repair leads to a decrease in the number of repaired components, since repair becomes less interesting compared to discard. If the probability of unsuccessful repair is lower at the higher echelon levels, then we expect more repairs to be performed at the higher echelon levels. If the probabilities decrease with an increasing echelon level, a second repair attempt may also be an option. In that case, we expect an increased number of repairs at the higher echelon levels, since the second repair attempt is necessarily at a higher echelon level.

## 5.2 No-fault-found

In practice, a certain fraction of the components that a repair shop receives, appears to function, a so-called no-fault-found. There are various reasons why a no-fault-found may occur: the diagnosis of the failure may be wrong, or a failure may occur due to the interaction of a component with the system; the component works if it is used in another system. After extensive testing, the component goes back in stock as-good-as-new.

Since a no-fault-found is not a real failure, it may seem reasonable to exclude it from the failure rate. However, the component may be tested extensively, possibly requiring a resource or shipment of the component to another location. As a result, the costs for a no-fault-found cannot be neglected. Still, a no-fault-found is different from a normal failure since the costs of finding a no-fault-found and performing a repair may differ and no decisions are required for subcomponents if a no-fault-found is detected, whereas they are required if a component is repaired by replacing a subcomponent. Since in general, finding a no-fault-found is less expensive than repairing a component (and its subcomponents), we expect that more repairs will be performed if we model the no-fault-found possibility.

## 5.3 Capacitated resources

In the basic LORA model, resources have infinite capacity, so at most one resource of the same type is required at each location. At Thales Nederland, this is a realistic assumption since the resource utilization rate is generally below 25%. At other companies, one resource may be insufficient to accommodate all demand and multiple resources may be required. This may happen especially if repairs are performed upstream in the repair network and the installed base is relatively large, consisting of hundreds of systems, which is quite common for, e.g., medical equipment. If multiple resources are required, the repair

strategy may change. We expect that less repairs will be performed, since acquiring the required resources is more expensive. Furthermore, we expect that repairs will be repaired at a lower echelon level. For example, if we need two resources at the central depot to handle all repair jobs and there are two intermediate depots, it may be more attractive to repair at the intermediate depots, thus avoiding transportation costs.

We do not take into account waiting times that arise from resource utilization since that only influences the availability and the amount of spare parts that may be required, which is not part of our model. Instead, we assume that we can specify in advance which resource utilization rate is acceptable (e.g., 70%). A drawback of this approach is that if the utilization rate is just slightly above the acceptable rate (e.g., 71% instead of 70%), we require two resources, whereas in practice, one resource may still be sufficient.

## 5.4 Multiple failure modes per component

In the basic LORA model, a single repair/discard decision is taken for each component, irrespective of the exact failure mode. Obviously, this does not need to be optimal: multiple failure modes may occur in one type of component, some being more serious than others, requiring different repair/discard decisions. To include multiple failure modes in the LORA model, we do not need to change the model. Instead, we can replace 'component' by 'failure mode', and change costs, parent-child relations between failure modes, and relations between resources and failure modes. However, the size of the problem grows, and we need more detailed information on failure modes as input for the model.

## 5.5 Outsourcing of repair

For some repair jobs, it may be cost-effective to outsource repair instead of carrying out repairs in the own service network. This is especially true if expensive resources are required to perform the repairs. Acquiring such a resource may only be cost-effective if it can be used for other repairs as well. Otherwise, it may be better to outsource the repair.

Including the outsourcing option in the minimum cost flow model is straightforward. We add the option 'outsource' at each location, or just at the central depot if all outsourced components should first be shipped to the central depot. The outsource option is specified similar to the discard option: no further decisions need to be taken for the component or its subcomponents. Furthermore, we have no resource costs, so only variable repair costs are to be taken into account (which may be higher than the variable repair costs for normal repairs). At Thales Nederland, we encountered the outsourcing decision for some components (outsource to the original equipment manufacturer).

# 6 Model formulation of extensions

In this section, we show how to model three of the extensions that we discussed in Section 5: unsuccessful repair, no-fault-found, and capacitated resources. As explained, the other two extensions do not require fundamental changes in the model.

## 6.1 Unsuccessful repair

As discussed in Section 5.1, a new decision needs to be taken for the unsuccessfully repaired components. In some circumstances, only discarding is possible, which we discuss in Section 6.1.1, whereas in other circumstances, a second repair attempt is possible (or even more attempts), which we discuss in Section 6.1.2. As a result of adding more options to the model, the model requires more nodes and arcs. We will clarify this throughout this section using a small example: we consider a component ($c_1$), having two subcomponents ($\Gamma_{c_1} = \{c_2, c_3\}$). We assume that component $c_1$ is repaired at location $l_2$. In Figure 3 in Section 4.1, we show the transformation node that results in the basic flow model; in the sections below, we show what modifications we make to this part of the flow model to incorporate a probability of unsuccessful repair.

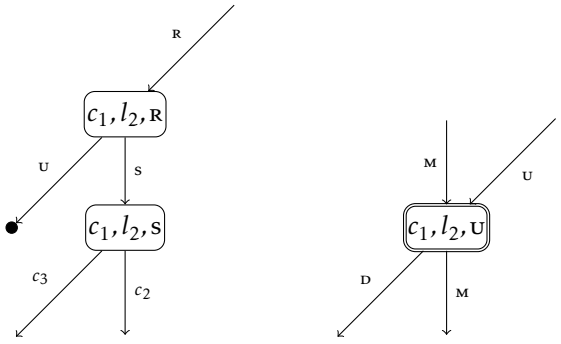
### 6.1.1 Discard only

The simplest case of incorporating a probability of unsuccessful repair is by assuming that after an unsuccessful repair, the component will be discarded at the location where the repair attempt was made. To this end, we add an additional transformation node, see Figure 4a, such that the incoming flow of components that are to be repaired, is split into two flows:

1. a flow of successfully repaired components (S), which goes to the already existing transformation node (renamed to $(c_1, l_2, \text{S})$; and
2. a flow of unsuccessfully repaired components (U) that will be discarded. This flow ends in a sink node.

The costs of unsuccessful repair and successful repair can be attached to the respective arcs.

If the costs of discard differ at the various echelon levels, it may be cost-effective to ship a component to a higher echelon level and discard it there. In that case, the flow representing the unsuccessfully repaired components (U) is sent to a decision node, see Figure 4b. At this node, there are the options to discard and to move the component.




(a) Discard at location of repair attempt

(b) Discard at higher echelon level

Figure 4: Unsuccessful repair

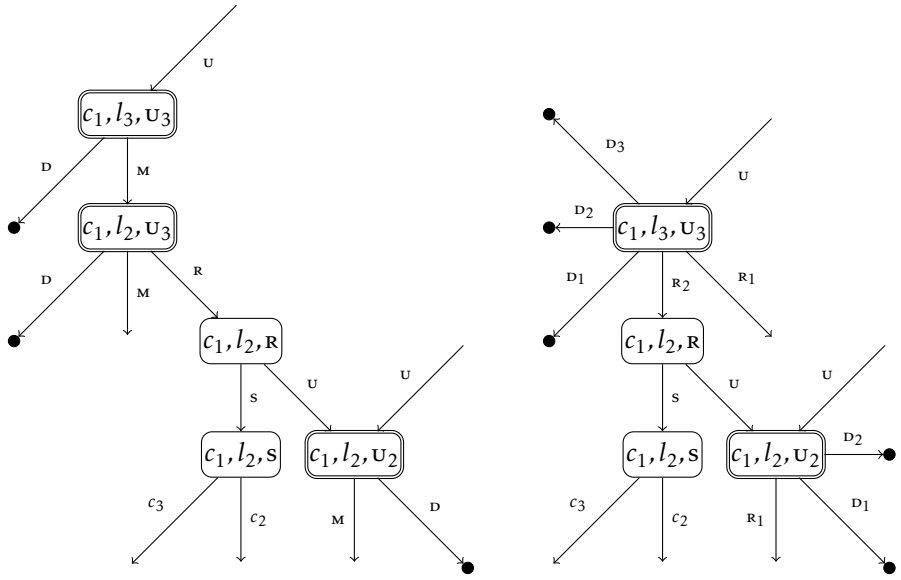
*Nodes*: A tuple (component, location, state), with state R representing *repair is chosen*, S representing *successfully repaired*, and U representing *unsuccessfully repaired. Arcs*: A component for which no decision has been taken yet, or a letter representing a state. In addition to states R, S, and U, state M represents *move is chosen* and D represents *discard is chosen.*

### 6.1.2 Second repair attempt

It may be useful to perform a second (and third, et cetera) repair attempt in some cases. In that case, we may specify a probability of successful repair that depends on the locations of all previous repair attempts (if any). However, we face two drawbacks. First, it is hard to specify all the required probabilities in practice. Second, the flows corresponding to unsuccessful repair at each location should be separated, which yields a very large model.

We propose a model in which only a few probabilities need to be specified. To this end, we assume that failures in one component can be ranked according to their complexity. If the probability of unsuccessful repair at one location is lower than at another location, all failures that will successfully be repaired at the latter location, will also be successfully repaired at the former location. As an example, we assume that $\Phi_{l_2} = \{l_3\}$ and that the probability of unsuccessful repair is 0.3 at location $l_3$ and 0.2 at location $l_2$. As a result of our assumptions, the fraction of the repairs that are successful at location $l_3$ (0.7) concern the least complicated failures, whereas the fraction that can be repaired at location $l_2$, but not at location $l_3$ (0.8-0.7=0.1) is more complicated, et cetera. If a first repair attempt is made at location $l_3$ and a second repair attempt is made at location $l_2$, then a fraction of $\frac{0.2}{0.3} = 0.67$ of the components that are unsuccessfully repaired at location $l_3$, are also unsuccessfully repaired at location $l_2$. Under these assumptions, a second repair attempt is useful only if the probability of unsuccessful repair is lower at the higher echelon level. Furthermore, it is only relevant to know the last location where a repair attempt has been performed; it does not matter whether any other repair attempts have been made.

We describe our new model for the example discussed above; the model is shown in Figure 5a. We start with a flow of components for which repair was unsuccessful at




(a) Standard version (b) Merged version

Figure 5: Unsuccessful repair: discard or second repair attempt
*Nodes*: A tuple (component, location, state), with state R representing *repair is chosen*, S representing *successfully repaired*, and U representing *unsuccessfully repaired*. A subscript indicates the location of the last unsuccessful repair. *Arcs*: A component for which no decision has been taken yet, or a letter representing a state. In addition to states R, S, and U, state M represents *move is chosen* and D represents *discard is chosen*. A subscript indicates where the repair or discard is to be performed.

location $l_3$ (30% of the failures, represented as $\mathrm{U}_3$ in the decision node). Such components can either be discarded (the flow goes to a sink node) or moved to the next higher echelon level. At location $l_2$ (the decision node $(c_1, l_2, \mathrm{U}_3)$), there are three possible decisions:

1. the component can be discarded,
2. the component can be moved to location $l_1$, or
3. a second repair attempt can be made.

A repair attempt is either successful (33% of the components) or unsuccessful (67%). In the latter case, the component cannot be distinguished from a component which was unsuccessfully repaired at location $l_2$ only (20%). Therefore, we can merge these two streams of components in the decision node $(c_1, l_2, \mathrm{U}_2)$.

We can simplify the representation by merging decision nodes such that after an unsuccessful repair, we immediately decide what the next repair/discard decision will be (at the current or a higher echelon level), see Figure 5b.

## 6.2 No-fault-found

Modelling the no-fault-found probability is similar to modelling the simplest case of a probability of unsuccessful repair (Section 6.1.1). In Figure 6, we show an example in

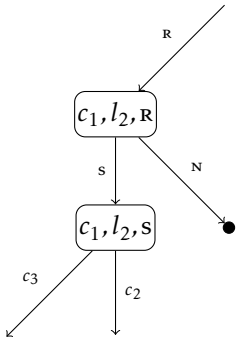



Figure 6: No-fault-found
*Nodes*: A tuple (component, location, state), with state R representing *repair is chosen* and S representing *successfully repaired*. *Arcs*: A component for which no decision has been taken yet, or a letter representing a state. In addition to states R and S, N represents *no-fault-found*.

which we assume that there is a probability of a no-fault-found (at location $l_2$). We add a transformation node, such that the incoming flow of components that are to be repaired, is split into two flows:

1. A flow of successfully repaired components (S), which goes to the already existing transformation node (renamed to $(c_1, l_2, \text{S})$).
2. A flow of components in which no failure is found , which terminates in a sink node, since the components are put back in stock and no additional decision is required.

The costs of a no-fault-found and a normal repair can be attached to the respective arcs.

## 6.3 Capacitated resources

To model capacitated resources, we change the side constraint that links arcs to resources in the minimum cost flow model (Constraint 5). The total flow over all arcs $(v, w) \in \Theta_{r,l}$ should be lower than the capacity per resource $r$ times the number of resources $r$ at location $l$. To this end, we define three parameters:

- $h_{c,r,d}$ is the demand in hours for resource $r$ per component $c$ for which decision $d$ is taken,
- $u_r$ is the annual number of hours that resource $r$ can be used (capacity of resource $r$ assuming a certain maximum utilization rate, e.g., 70%), and
- $g_{v,w,r}$ is the demand in hours for resource $r$ if there is a flow of one on arc $(v, w)$: $g_{v,w,r} := h_{c,r,d}$ if arc $(v, w)$ represents decision $d$ for component $c$ at a certain location.

$Y_{r,l}$ is now a general integer variable (instead of a binary) and we change Constraint 5 to:

$$\sum_{(v,w)\in\Theta_{r,l}} g_{v,w,r} \cdot F_{v,w} \leq u_r \cdot Y_{r,l} \ , \forall r \in R \ , \forall l \in L \tag{8}$$

# 7 Computational experiments

In this section, we perform an extensive numerical experiment in which we examine how each of the three extensions that we discussed in Section 6 influence:

1. the total costs,
2. the repair strategies (LORA decisions), and
3. the optimization times.

The main goal of our experiment is to examine whether the extensions need to be incorporated in order to get accurate, practically useful results. If repair strategies do not change drastically, not incorporating the extensions keeps the model and required input data simple. If costs do change, it may be possible to find a good repair strategy using a simple model and to adapt the costs afterwards. Only if repair strategies change, the extensions need to be incorporated when using the model in practice.

We do not perform tests for the other two extensions that we discussed in Section 5. Multiple failure modes in one component can occur in various ways and require setting many different parameters. We believe that setting these parameters in a practically relevant way requires company input. Currently, this input is not available to us. The outsource option can easily be incorporated and there is no reason not to do this in practice (problem size and therefore optimization time do not increase much).

We perform tests with a probability of unsuccessful repair for the case that discard is allowed only after the unsuccessful repair and for the case that a second repair attempt is possible. In the former case, we assume that discard is necessarily performed at the location of the repair attempt. Furthermore, we assume in our tests that the costs of an unsuccessful repair, a successful repair, and finding a no-fault-found are equal. Finally, we assume that the failure rates of the LRUs are not changed if an extension is incorporated. This means that the number of 'actual failures' decreases if we incorporate a no-fault-found probability.

In Section 7.1, we discuss the basic scenarios that we use in our tests. We perform tests for each of the extensions, one by one. In Sections 7.2 to 7.4, we give for each of the extensions the additional parameters that we use and the test results. We solve problem instances using the CPLEX callable library version 11 (with default settings), running under windows XP, service pack 2, on an Intel Centrino Duo, 2 GHz with 2 GB RAM.

## 7.1 Basic scenarios

For our numerical experiment, we generate problem instances using a modified version of the generator that was presented by Basten et al. (2008). We consider smaller repair networks, since incorporating capacitated resources leads to a larger number of decision

variables and thus to higher memory requirements: the largest networks become to big to be handled by CPLEX. Furthermore, we consider symmetrical repair networks only, so that we can focus more easily on the effects of the extensions; Basten et al. (2008) also consider asymmetrical repair networks. We outline the generator below and we describe it in detail in Appendix A.

In all problem instances, we use a three-indenture product structure consisting of 25 LRUs, 125 SRUs, and 625 parts. The three-echelon repair network consists of 2 or 5 intermediate depots and 2 or 5 operating sites per intermediate depot (at most 25 operating sites in total).

There are 10 or 25 resources. 70% of the components require no resource, 20% require one resource, and 10% require two resources. We also perform tests with these percentages being 25%, 50%, and 25%, respectively.

Component prices are in the range [1,000; 100,000] and resource prices are in the range [10,000; 1,000,000]. Failure rates of LRUs are in the range [0.01; 1] and we assume that resources are required for repair only (not for discard or move).

Summarizing, there are four experimental factors, namely the number of intermediate depots, the number of operating sites, the number of resources, and the number of resources per component. This makes $2^4 = 16$ settings. Since we generate ten problem instances at random for each of the parameter settings, there are 160 basic scenarios.

## 7.2 Probability of unsuccessful repair

For the probability of unsuccessful repair, we distinguish nine settings. We give the values that we use for the probabilities in Table 1. They depend on the indenture level of the component and the echelon level of the location. In settings 1 to 3, the probabilities are equal at all echelon levels and at all indenture levels. Under the assumptions given in Section 6.1.2, having equal probabilities of unsuccessful repair means that no additional repairs can be performed at the higher echelon levels. Therefore, performing tests using these settings makes no sense for the tests with a 'second repair attempt at higher echelon level'. In settings 4 to 6, the probabilities are equal at all indenture levels. However, the probabilities are lower at the higher echelon levels, since in general, more specialized engineers and equipment will be available there. In settings 7 to 9, the probabilities still vary over the echelon levels. In addition, the lower the indenture level of the component (e.g., LRU), the lower the probability of unsuccessful repair. Since these components are relatively expensive, it is worthwhile to invest time and money if the failure is hard to repair. Besides, it is more probable that a part of the LRU can be replaced, thus repairing the LRU, whereas this is often not possible for the higher indenture level components (e.g., parts).

| | Echelon 1 | | | Echelon 2 | | | Echelon 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Setting | LRU | SRU | part | LRU | SRU | part | LRU | SRU | part |
| 1 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 |
| 2 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 |
| 3 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 |
| 4 | 0.09 | 0.09 | 0.09 | 0.06 | 0.06 | 0.06 | 0.03 | 0.03 | 0.03 |
| 5 | 0.18 | 0.18 | 0.18 | 0.12 | 0.12 | 0.12 | 0.06 | 0.06 | 0.06 |
| 6 | 0.27 | 0.27 | 0.27 | 0.18 | 0.18 | 0.18 | 0.09 | 0.09 | 0.09 |
| 7 | 0.06 | 0.075 | 0.09 | 0.045 | 0.06 | 0.075 | 0.03 | 0.045 | 0.06 |
| 8 | 0.12 | 0.15 | 0.18 | 0.09 | 0.12 | 0.15 | 0.06 | 0.09 | 0.12 |
| 9 | 0.18 | 0.225 | 0.27 | 0.135 | 0.18 | 0.225 | 0.09 | 0.135 | 0.18 |

Table 1: Parameters: probability of unsuccessful repair

| | Variable costs | | | Resource costs | | | Total |
|---|---|---|---|---|---|---|---|
| | Discard | Repair | Move | Ech. 1 | Ech. 2 | Ech. 3 | costs |
| Basic scenario | 553 | 6,732 | 1,670 | 64 | 2,513 | 1,221 | 12,753 |
| Settings 1 – 9 | 12,004 | 7,176 | 2,513 | 22 | 1,767 | 1,335 | 24,817 |
| Settings 1 – 3 | 13,888 | 6,381 | 1,635 | 64 | 2,502 | 1,061 | 25,532 |
| Settings 4 – 9 | 11,062 | 7,574 | 2,952 | 0 | 1,399 | 1,472 | 24,456 |

Table 2: Costs (×1,000), resulting from using a probability of unsuccesful repair (immediate discard)

Since there are 160 scenarios for each parameter setting, there are $9 \cdot 160 = 1{,}440$ problem instances in total for the tests 'discard at location of repair attempt', and $6 \cdot 160 = 960$ problem instances for the tests 'second repair attempt at a higher echelon level'.

### 7.2.1 Discard at location of repair attempt

Incorporating a probability of unsuccessful repair leads to almost a doubling of the costs on average: 24.8 million versus 12.8 million. The maximum cost increase is 256%. The reason is that many components cannot be repaired successfully and need to be discarded. However, discarding components is very expensive, since that means that replacements need to be purchased.

In Table 2 (first two rows), we see that not only the solution value changes, the repair strategy changes as well: resources are located differently. To understand what happens, we have to look in more detail to the results.

Using settings 1 to 3 (equal probability of successful repair at all echelon levels), less repairs are performed and less components are moved than in the basic scenario (compare rows 1 and 3). Since a probability of unsuccessful repair means that in some cases, no subcomponents are replaced, there are simply less components in total that

require a repair or move action. This decrease in demand further means that in some cases, it is not worthwhile to acquire a resource, as can be seen in Table 2 (upstream in the network); it can be sensible to discard a component immediately if a component is not too expensive, its repair requires a resource, However, we conclude that the change in repair strategy is minor.

Using settings 4 to 9, the repair strategy changes drastically compared to the basic scenario (compare rows 1 and 4). The variable move costs and the resource costs at central depot increase, whereas the resource costs at the lower echelon levels decrease. The reason is that is worthwhile to ship components to the central depot, since that leads to a low probability of discarding the component. Also, the repair costs increase, which means that a repair attempt is made on more components (instead of discard). We cannot explain why this happens.

The computation time increases from 0.5 seconds to 1.3 seconds on average if the probability of unsuccessful repair is incorporated (less than 30 seconds at maximum).

### 7.2.2 Second repair attempt at a higher echelon level

The inclusion of a second repair attempt at a higher echelon level yields a cost increase of 71% on average compared with ignoring unsuccessful repair: 21.8 million versus 12.8 million. The maximum cost increase is 179%. We have seen in Table 2 (fourth row) that if we allow for discard at the location of the repair attempt only, costs increase to 24.5 million. This means that if the probabilities differ at the various echelon levels, it is worthwhile to allow for a second repair attempt. The optimization time will not be restrictive. It increases from 0.5 seconds to 5.7 seconds on average (maximum optimization time is under 56 seconds).

Table 3 gives more detailed results. Compared with the results from allowing for discard only (Table 2, fourth row), the discard and move costs increase less, whereas the variable repair costs increase more. The resource costs decrease far less at echelon levels 1 and 2 and increase more at the central depot. So, the additional option of a second repair attempt leads to more components being repaired. Instead of shipping and repairing components upstream, now a first repair attempt is made on many components at the lower echelon levels, and a second (and third) repair attempt is made upstream. Clearly, if a second repair attempt is possible in practice, this extension should be modelled to get realistic results.

| | Variable costs | | | Resource costs | | | Total |
|---|---|---|---|---|---|---|---|
| | Discard | Repair | Move | Ech. 1 | Ech. 2 | Ech. 3 | costs |
| Basic scenario | 553 | 6,732 | 1,670 | 64 | 2,513 | 1,221 | 12,753 |
| Settings 4 – 9 | 7,549 | 8,564 | 2,322 | 14 | 1,641 | 1,689 | 21,780 |

Table 3: Costs (×1,000), resulting from using a probability of unsuccesful repair (repair/-discard at higher echelon)

| Setting | LRU | SRU |
|---|---|---|
| 1 | 0.06 | 0.06 |
| 2 | 0.12 | 0.12 |
| 3 | 0.18 | 0.18 |
| 4 | 0.09 | 0.03 |
| 5 | 0.18 | 0.06 |
| 6 | 0.27 | 0.09 |

Table 4: Parameters: no-fault-found probability

## 7.3 No-fault-found probability

To examine the impact of the no-fault-found probability on the results, we perform tests with six different settings, see Table 4. In our tests, finding a no-fault-found is equally expensive as performing a repair; the only difference is that a no-fault-found means that no decisions need to be taken for subcomponents. Therefore, setting a no-fault-found probability impacts only components that have subcomponents, i.e., not parts. We set the no-fault-found probability for each component equal at all echelon levels. Furthermore, they are equal at both indenture levels in settings 1 to 3. In settings 4 to 6, the probability for the LRUs is higher than that for the SRUs, since LRUs are technically more complex. Besides, LRUs are always replaced in the field, whereas SRUs are typically replaced in a repair shop. This means that the probability of an incorrect replacement (another component failed) is generally higher for LRUs than it is for SRUs.

In general, incorporating a no-fault-found probability leads to lower costs (-4% on average) and an increase in the computation time (0.5 seconds versus 1.6 seconds). The maximum cost reduction that can be achieved is over 12%.

We see in more detail how costs change in Table 5. Less resources are located in the network, the discard costs increase, and the (variable) repair and move costs decrease. The reasoning to explain the results is as follows. Some of the LRUs that are repaired turn out to be a no-fault-found, which means that its SRUs need not be repaired or discarded. The same holds for SRUs and parts, respectively, which means that the total demand for repair and discard actions decreases in the LORA problem. For some resources, the demand gets so low that it is cost-effective to discard all components that require that

| | Variable costs | | | Resource costs | | | Total |
|---|---|---|---|---|---|---|---|
| | Discard | Repair | Move | Ech. 1 | Ech. 2 | Ech. 3 | costs |
| Basic scenario | 553 | 6,732 | 1,670 | 64 | 2,513 | 1,221 | 12,753 |
| Settings 1 – 6 | 566 | 6,402 | 1,644 | 67 | 2,455 | 1,163 | 12,297 |

Table 5: Costs (×1,000), resulting from using a no-fault-found probability

| | Costs (×1,000) | Optimization time (seconds) |
|---|---|---|
| Basic scenario ($RR = 1$) | 12,753 | 0.5 |
| $RR = 2$ | 13,383 | 8.1 |
| $RR = 4$ | 15,023 | 34.6 |

Table 6: Results for various settings for the capacitated resources

resource, instead of locating the resource and repairing the components. However, the change in the repair strategy is small, which means that modelling the no-fault-found probability may not be necessary in practice.

## 7.4 Capacitated resources

To test the extension to capacitated resources, we need to set two sets of parameters:

- the demand for a resource that results from repairing one component, and
- the annual capacity of the resource.

For simplicity, we assume that the repair of each type of component requires the same amount of time. Then, notice that for each resource, the total demand at any location cannot be higher than the total demand at the central depot if all components are repaired there. We design our experiments such that it holds for all resources that if all components are repaired at the central depot, we require $RR$ resources at the central depot. We perform tests with $RR = 2$ and $RR = 4$. Notice that $RR = 1$ effectively represents uncapacitated resources.

The cost difference due to assuming capacitated resource compared with assuming uncapacitated resources is quite large, as can be seen in Table 6. If $RR = 4$, it is 18% on average and 59% at maximum. The optimization time increases drastically as well; it is almost half an hour at maximum (compared with 2.3 seconds for the basic scenario).

If resources are capacitated, the repair option gets more expensive compared with the discard option. Therefore, less repairs are performed: Table 7 shows that the discard costs increase and the variable repair costs decrease. The total costs for resources increase as well, because sometimes multiple resources of the same type are required. If this happens at the central depot, it is sometimes possible to perform repairs at a lower echelon level without increasing the total number of required resources. However, it does reduce the

| | Variable costs | | | Resource costs | | | Total |
|---|---|---|---|---|---|---|---|
| | Discard | Repair | Move | Ech. 1 | Ech. 2 | Ech. 3 | costs |
| Basic scenario | 553 | 6,732 | 1,670 | 64 | 2,513 | 1,221 | 12,753 |
| $RR \in \{2, 4\}$ | 1,477 | 5,935 | 1,422 | 490 | 4,129 | 748 | 14,203 |

Table 7: Costs (×1,000), resulting from using capacitated resources

| Parameter | Setting | Cost difference |
|---|---|---|
| # intermediate depots | 2 | 16.9% |
| | 5 | 10.3% |
| # operating sites per intermediate depot | 2 | 16.5% |
| | 5 | 10.7% |
| # resources | 10 | 11.6% |
| | 25 | 15.6% |

Table 8: Cost difference due to using capacitated resources for various parameter settings
The cost difference is the average over all problem instances of: the total costs incorporating capacitated resources minus the total costs neglecting the resource capacities, divided by the total costs neglecting the resource capacities.

move costs. Therefore, we see that resources are located at lower echelon levels and that move costs decrease. Since the resource locations change substantially if resources become capacitated, this extension cannot be ignored in practice if resource capacities are tight.

As shown in Table 8, the cost difference between assuming capacitated and uncapacitated resources decreases if the number of intermediate depots increases. The reason is that with five intermediate depots, additional resources may be required only if repairs are performed at the central depot. The reasoning is as follows. If there are two intermediate depots and $RR = 4$, then performing repairs at the intermediate depots may mean that two resources per intermediate depot are required. However, if there are five intermediate depots, then locating one resource at each of the intermediate depots is always sufficient, due to the way we construct our problem instances (see our definition of $RR$).

If the number of operating sites per intermediate depot decreases or the number of resources increases, the cost difference increases as well. This makes sense, since the resource costs as a percentage of the total costs increase, and Table 7 shows that the overall cost increase results mainly from an increase in the resource costs.

# 8 Conclusions and recommendations for further research

We discussed the following extensions to the LORA model:

- A probability of unsuccessful repair
- A no-fault-found probability
- Capacitated resources
- Multiple failure modes per component
- Outsourcing of repair

We tested the first three extensions and found that incorporating them leads to a substantial change in the costs. However, the change in repair strategy is minor when incorporating a no-fault-found probability or a probability of unsuccessful repair that is equal at all echelon levels. As a result, it may be possible to find a repair strategy that is close to optimal using a simple model and to adapt the costs afterwards. If the probability of unsuccessful repair decreases with an increasing echelon level or if capacitated resources are modelled, the repair strategies change drastically. In the former case, the resource costs at central depot increase, wherease they decrease at the lower echelon levels. When modelling capacitated resources, more components are discarded and resources are located at lower echelon levels. Furthermore, the optimization time increases strongly.

Although we modelled the most important extensions that may be required for specific business situations, other extensions may be required as well. For example, we assume that we know which resource we need to repair a component. In practice, this is not necessarily known: we can face the decision to buy a rather cheap resource that can be used for the repair of a single component or a more expensive resource that can be used to repair multiple components (see, e.g., Alfredsson, 1997). A similar decision is to buy a cheap resource that leads to a relatively high probability of unsuccessful repair or a more expensive resource that leads to a lower probability of unsuccessful repair. These extensions may be included in the model by adapting the constraint that relates the repair option to the resources. Modelling and testing these extensions and testing the extension to multiple failure modes, based on company data, is interesting future research.

# Acknowledgments

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

# Appendix A

We describe the random number generator that we use for our numerical experiments. We explain how we generate repair network structures, product structures, failure rates

and cost factors. We give a summary of all parameter settings in Table 9.

The three-echelon repair network consists of 2 or 5 intermediate depots, each connected to 2 or 5 operating sites. As a result, there are 4, 10 or 25 operating sites in total.

In all tests, the system structure consists of 25 components at the first indenture level (subsystems), 125 at the second level and 625 at the third level. Every second and third indenture component is randomly attached to a lower indenture component with equal probability per parent component. As a result, it may occur that a first indenture component has zero subcomponents (which also occurs in practice, e.g., at Thales Nederland). The annual demand per subsystem is drawn from a uniform distribution on the interval $[0.01; 1]$. The annual demand for a higher indenture component is a certain fraction of the demand for its parent component. The fraction of demand for the parent component that is due to a failure in the child component is drawn from a uniform distribution on the interval $\left[\frac{0.5}{\text{number of children}}; \frac{1.25}{\text{number of children}}\right]$, with a maximum of 1. Summing the fractions of all children leads to a value between 0.5 and 1.25. If this summation is lower than 1, then a certain fraction of the failures in the parent can be repaired directly, without replacement of a child component. If the summation is higher than 1, then a certain fraction of the failures in the parent leads to the replacement of more than one child component.

For each component, we draw a net price, excluding the costs of child components, from a shifted exponential distribution with shift factor 1,000 and rate parameter $\lambda = 7/(100{,}000 - 1{,}000)$. In this way, on average 1‰ of the components get a price higher than €100,000. We draw a new price for these components to avoid odd problem instances. The reason for our choice is that most systems have a large diversity of items in price, but there are considerably more cheap items than expensive items. The cheapest items (in our case items with a price below €1,000) are usually omitted from a regular LORA, because they are discarded by default.

Using these prices, we calculate the variable costs as follows:

- Repair costs as a fraction of the net component price are drawn from a uniform distribution on the interval $[0.1; 0.4]$.
- We recursively add the price of each child to its parent to get the gross item price. We do this after calculating the repair price, since repair of a parent means replacement of the child that was defective and taking a decision for the child, thus incurring costs for the child. Discarding or moving a parent does not lead to a decision for the child components, and thus not to incurring costs for the child components either.
- The discard costs as a fraction of the gross item price, including children, are drawn from a uniform distribution on the interval $[0.75; 1.25]$. 100% would be just the costs of replacing a defective component by a new one. However, there may be additional

disposal costs to satisfy legal requirements for waste processing. On the other hand, some parts of a defective component may be recycled or re-used. Therefore, we vary the disposal costs around the price of a new component.

- The move costs are always 1% of the gross item price.

We perform tests with 10 and 25 resources. The price of each resource is drawn from a shifted exponential distribution with shift factor 10,000 and rate parameter $\lambda = 7/(1{,}000{,}000 - 10{,}000)$ in the same way as described above for the prices of the components. As a result, prices vary between €10,000 and €1,000,000. We randomly assign components to resources, such that the percentage of components that needs 0, 1 and 2 resources are 70%, 20% and 10%, respectively. To be short, we write '0.7–0.2–0.1'. We also perform tests with settings '0.25–0.5–0.25'.

We also need to take spare parts holding costs into account. The reason is that in the LORA, high resource costs typically cause that repairs are performed upstream in the repair network, whereas high transportation costs cause downstream repairs, close to the installed base. Since in a high tech environment resources tend to be expensive, repairs of components that require resources are generally performed at the central depot. However, this leads to high lead times and thus to high spare parts costs. Taking into account the spare parts costs in the LORA is not straightforward. In Section 2, we discussed some literature in which this joint problem is considered. In general, the problem is that LORA requires spare parts holding costs per component, whereas typically, a multi-item approach (or system-approach) is used in practice to solve the spare parts stocking problem, e.g., the VARI-METRIC type approaches (see, for example, Sherbrooke, 2004).

We use a simple item-approach here. It is based on the difference in lead times that result from different LORA decisions. We assume that repair at the system location or at the intermediate depot takes a month, whereas repair at the central depot takes three months. Discard (and buying a new item) has a lead time of half a year. Moving a component to a higher echelon leads to an additional lead time of half a month. Using these lead times and a safety factor of 2, we estimate spare parts inventory as being the safety factor times the lead time demand. To compute the corresponding holding costs, we use an inventory carrying charge of 30%. We tested other approaches and safety factors, but found that this approach generally leads to reasonable results.

# References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows. Theory, algorithms, and applications.* Prentice Hall, Englewood Cliffs (NJ).

Alfredsson, P. (1997). Optimization of multi-echelon repairable item inventory systems

Table 9: Parameter settings

| Parameter | Value(s) or Range |
|---|---|
| # Central depots | 1 |
| # Intermediate depots | 2 & 5 |
| # Operating sites per intermediate depot | 2 & 5 |
| # LRUs | 25 |
| # SRUs | 125 |
| # Parts | 625 |
| Demand per LRU | $[0.01; 1]$ |
| Fraction of demand for parent due to child | $\left[\frac{0.5}{\text{number of children}}; \frac{1.25}{\text{number of children}}\right]$ |
| Price of component | [1,000; 100,000] |
| Repair costs (fraction net price) | $[0.1; 0.4]$ |
| Discard costs (fraction gross price) | $[0.75; 1.25]$ |
| Move costs (fraction gross price) | 0.01 |
| Resource costs | [10,000; 1,000,000] |
| # Resources | 10 & 25 |
| Fraction of components requiring 0–1–2 resources | 0.7–0.2–0.1 & 0.25–0.5–0.25 |
| Repair lead time at echelon level 1 | 1 month |
| Repair lead time at echelon level 2 | 1 month |
| Repair lead time at echelon level 3 | 3 months |
| Move lead time between echelons (1 & 2, 2 & 3) | 0.5 months |
| Discard lead time | 6 months |
| Safety factor | 2 |
| Holding costs | 0.3 |

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. J. I., Kutanoglu, E., Van der Heijden, M. C., and Schutten, J. M. J. (2009a). An optimal approach for the joint problem of level of repair analysis and spare parts stocking. BETA working paper 298. Submitted for publication.

Basten, R. J. I., Schutten, J. M. J., and van der Heijden, M. C. (2009b). 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. (2008). A minimum cost flow model for level of repair analysis. BETA working paper 254. Submitted for publication.

Basten, R. J. I., Van der Heijden, M. C., and Schutten, J. M. J. (2009c). An iterative method for the simultaneous optimization of repair decisions and spare parts stocks. BETA working paper 295. Submitted for publication.

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.

Daskin, M. S. (1995). *Network and discrete location. Models, algorithms, and applications*. John Wiley & Sons, New York (NY).

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

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