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

February 17, 2010

1 Introduction

We propose a method that can be used when deciding on how to maintain capital goods, given a product design and the layout of a repair network. Capital goods are physical systems that are used to produce products or services. They are expensive and technically complex and have high downtime costs. Examples are manufacturing equipment, defense systems, and medical devices.

Due to the high downtime (unavailability) costs of capital goods, a failed component will usually be repaired by replacement by a spare part. In the defense industry, the components that are replaced are called LRU’s or line replaceable units. Many of these components are expensive as well and are therefore repaired. These repairs are typically performed by replacement of a subcomponent. Since this is usually done in a repair shop, these subcomponents are called SRU’s or shop replaceable units. Instead, it may also be decided to discard the LRU and buy a new one. 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. Such resources may include test, repair, and transportation equipment, but also one time training of service engineers. Spare parts need to be located in the repair network as well, in order to enable quick repairs of both the installed base (spare LRU’s) and of LRU’s and SRU’s (spare SRUs and parts, respectively).

The spare parts stocking analysis is conducted to decide which components to put on stock at which location(s) in a repair network in which quantity, such that a target availability of the capital goods is achieved against minimum spare parts investment costs. A well-known method to conduct this analysis is VARI-METRIC, which uses as input:

Figure 1: Product structure

Figure 2: Repair network
• data about the product structure, such as relations between components and their sub-components, cost data, and failure rates; and

• data about the network structure, such as relations between locations and their supplying locations, and lead times for repair, transportation, and resupply of newly purchased components.

Furthermore, the following repair/discard decisions are an input:

(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.

Given these decisions, VARI-METRIC generates a trade-off curve of spares investment costs versus optimal mean LRU backorder levels at the operating sites, a backorder meaning that a spare part is requested, but not available. As an approximation, the number of backorders equals the number of capital goods that are unavailable waiting for spares. Therefore, if we use the term availability, we mean the supply availability, which is defined as \( \frac{MTBM}{MTBM + MTWS} \), with MTBM being the mean time between (corrective) maintenance and MTWS being the mean time waiting for spares.

The level of repair analysis (LORA) can be used to make the above-mentioned repair/discard decisions, which are input for the spare parts stocking analysis, in conjunction with the decisions:

(3) at which locations to deploy resources.

The LORA is typically modeled as a deterministic integer optimization problem (see, e.g., Basten et al., 2009a, 2008). As a result, it is not possible to consider the downtime waiting for spares and the availability of the installed base. Since the spare parts costs can be included only in a rudimentary way, usually only the other relevant parts of the life cycle costs (LCC) are considered, consisting of both costs that are variable in the number of failures and fixed costs. Variable costs may include transportation costs, working hours of service engineers, and usage of bulk items; fixed costs are due to, for example, training of service engineers and depreciation of repair equipment.

Performing the LORA first and next the spare parts stocking analysis, the so-called sequential approach, may lead to a solution that is not optimal overall (LORA costs and spare parts investment). The reason for this is as follows. 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. On the other hand, repairing centrally leads to higher transportation costs. Since the transportation costs are generally low compared to costs for resources in a high-tech environment, the LORA often recommends to perform repairs at the central depot. The LORA neglects the fact that this leads to a repair lead time (including transportation lead times) being higher than if 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).

Basten et al. (2009b), which we discuss in the next section, develop an iterative method to solve the joint problem of LORA and spare parts stocking. Using this method, they show that a lot can be gained by solving the joint problem instead of performing a LORA first and next a spare parts stocking analysis. However, they cannot give an indication of the quality of their solutions. We propose a method that generates a VARI-METRIC-like trade-off curve of spares investment costs versus backorders. The points on the curve represent optimal backorder levels for a given investment level. To guarantee optimal backorder levels, we have to consider many possible solutions, which leads to high computation times. However, we show how to decompose the problem into smaller subproblems that can be solved in a reasonable amount of time.

In Section 2, we discuss the related literature and the contribution of our paper in more detail. We outline our model for the joint problem of LORA and spare parts stocking in Section 3. We discuss VARI-METRIC in Section 4, since our method is inspired by its logic. In Section 5, we present our algorithm and in Section 6, we show the results of our numerical experiments. In this section, we also present the results of a case study that we performed at Thales Nederland.
a manufacturer of naval sensors and naval command and control systems. In Section 7, we give conclusions and recommendations for further research.

2 Literature review

We discuss the literature on spare parts stocking, LORA, and the joint problem of LORA and spare parts stocking in Sections 2.1 to 2.3, respectively. Next, we show our contribution in Section 2.4.

2.1 Spare parts stocking

A vast amount of literature exists on the multi-item spare parts stocking problem. Lines of research can be distinguished based on their focus on repairable or consumable components, and on fast moving, inexpensive components or slow moving, expensive components. In the context of this research, we are interested in expensive, slow moving, repairable components. The work of Sherbrooke (1968) is generally seen as the seminal paper in this field. He developed the METRIC model (Multi-Echelon Technique for Recoverable Item Control), which is the basis for a huge stream of METRIC type models. The initial model can be used for single-indenture items only. Muckstadt (1973) developed the first multi-echelon, multi-indenture model, called MOD-METRIC. The development of the VARI-METRIC models (Slay, 1984; Graves, 1985; Sherbrooke, 1986) has been another important step forward; the relations between various echelon levels and indenture levels are more accurate in these models. These relations can also be calculated exactly (Graves, 1985; Rustenburg et al., 2003), but this is computationally intensive. The METRIC type models aim to find the most cost effective allocation of spare parts in a network that achieves a target availability of the installed base. These models can be solved using a Lagrangian approach or using a marginal analysis approach. The marginal analysis approach leads to a trade-off curve of spare parts investment costs versus backorders, a so-called EBO-curve. Since we use such curves in our algorithm, we discuss the marginal analysis approach in more detail in Section 4. We refer to Sherbrooke (2004) and Muckstadt (2005) for an extensive overview of the METRIC type models.

2.2 Level of repair analysis

We found six papers in which a model is presented for the LORA problem (Barros, 1998; Barros and Riley, 2001; Saranga and Dinesh Kumar, 2006; Basten et al., 2009a, 2008; Brick and Uchoa, 2009). In all these papers, the model is an integer linear programming (ILP) model.

The key problem with three of these models is that they are very restrictive in the component-resource relations. Barros (1998) and Barros and Riley (2001) assume that there are as many resources as there are indenture levels, each resource being used by all components at one indenture level, and Saranga and Dinesh Kumar (2006) assume that each component uses its own unique resource. The other three models allow for multiple components sharing resources and for components requiring multiple resources simultaneously. A problem of the model of Brick and Uchoa (2009) is that only one echelon level and two indenture levels can be modelled.

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. (2009b). In Section 2.3.1, we discuss the algorithm of Alfredsson (1997) in more depth since our algorithm is inspired by his. In Section 2.3.2, we discuss the work by Basten et al. (2009b). We use the same model as they do.

2.3.1 Marginal analysis approach.

Alfredsson (1997) assumes a single-indenture product structure and a two-echelon repair network. Each component requires one specific tester (resource), which is required by one component only. Furthermore, one multi-tester exists. This multi-tester can be used for the repair
of one component 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 component, the original specific resource for that component is not used anymore. Resources have finite capacity, which means that multiple resources of the same type may be required at one location. System downtime includes waiting times for resources, repair times, and waiting times for spares. The model is a non-linear integer programming model.

The author iteratively decomposes the problem in two ways (we refer to them as decomposition type 1 and 2, respectively). For the subproblems that result from decomposing the problem, he constructs an EBO-curve using a metric-like marginal analysis method. The two types of decomposing the problem and combining the results are:

1. Decomposition of the problem into one subproblem for each resource. Splitting per resource is allowed since the location of one resource does not influence the location of other resources. This results from assuming one indenture level only and the assumption that each component (LRU) requires exactly one resource to be repaired. The author further decomposes the problem into one subproblem for each LRU, as in the standard metric type models. In Section 4, we explain the marginal analysis approach that can be used to merge the results after solving each subproblem.

2. Decomposition of the problem into subproblems by fixating decision variables. For example, each resource in his model needs to be at exactly one of the two echelon levels. For each resource, he solves a subproblem in which the resource is at the lower echelon, and a subproblem in which the resource is at the higher echelon. In Section 4, we explain how to find the lower envelope over various EBO-curves, which can be used to merge the results of the two subproblems.

2.3.2 Iterative approach.

Basten et al. (2009b) use a model that is more general than the model of Alfredsson (1997) with respect to the component-resource relations that can be modelled. These relations are the same as in the LORA model of Basten et al. (2008), so multiple components may require the same resource in order to be repaired and components may require multiple resources simultaneously. Furthermore, Basten et al. (2009b) have no limit on the number of echelon and indenture levels. However, the model does not incorporate capacitated resources or the choice between multiple resources. The authors iteratively solve the model. The LORA is performed, using the model of Basten et al. (2008), and then VARI-METRIC is used to conduct the spare part stocking analysis. The key issue is the feedback mechanism: the total spare parts holding costs that result in the complete network are decomposed into spare parts holding costs per component and added to the variable costs in the LORA inputs. After a number of iterations, the LORA incorporates an estimate of the spare parts holding costs that will result from each possible repair/discard decision. As a result, the LORA will choose a solution that leads to low total costs (LORA and spare parts holding costs). The authors show that they achieve cost reductions of over 3% on average and over 35% at maximum compared to solving the problem sequentially. However, they cannot give an indication of the quality of their solutions, since the optimal solution is unknown and there is no good lower bound.

2.4 Contribution

The contribution of our paper is two-fold:

- Compared with the model of Alfredsson (1997), the model that we use is more general in certain aspects. In fact, we use the model of Basten et al. (2009b), so we allow for components sharing resources and for components requiring multiple resources simultaneously. Furthermore, we have no restriction on the number of indenture and echelon levels. However, we do not incorporate the waiting time for resources and the choice between two resources.

- For the joint problem of LORA and spare parts stocking, our method finds optimal back-order levels for a given investment level. Because of the optimality of our solutions, our
method can also be used as a benchmark for other methods, which is what we do in Section 6 for the iterative method of Basten et al. (2009b).

3 Model

In this section, we outline the model that we use. In Section 3.1, we give the problem description including the assumptions that we make. In Section 3.2, we present the mathematical model formulation. Finally, we make two further assumptions in Section 3.3, which ease the presentation in the remainder of this paper and decrease the problem size. It is, however, not a problem for our algorithm to relax these assumptions.

3.1 Problem description

We model the joint problem of LORA and spare parts stocking. Our goal is to achieve the lowest possible \( \text{LCC} \), subject to a constraint on the availability. The \( \text{LCC} \) include the variable and fixed LORA costs, as mentioned in Section 1, and the spare parts holding costs.

We assume that resources are uncapacitated and that the repair network is symmetrical (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). The latter assumption is not a critical assumption for our algorithm, but it reduces the problem size. To further reduce the problem size, we make the same decisions at all locations at one echelon level for each component and resource. In a symmetrical repair network, this is an optimal strategy, except that the overshoot increases (the difference between the achieved availability and the target availability, see Section 4). This is due to necessarily stocking a spare part at each location at one echelon level if stocking a spare part at only one or a few of these locations would suffice to reach the target availability.

The key decision to make is to decide whether a component will be discarded or repaired upon failure, and at which echelon level to perform the repair or discard. We model these decisions using three options at each echelon level for each component: discard, repair, and move to the next higher echelon level (if such a level exists). Choosing any of these three options for a component means that the required resources (if any) should be installed at that echelon level as well. Furthermore, spare parts may be required at the location where a component is replaced or at a higher echelon level, in order to reduce system downtime.

We assume that:

1. components fail according to a Poisson process with constant rate;
2. a failure in a component is caused by a failure in at most one subcomponent;
3. 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 the parent-location at the next higher echelon level;
4. for each component at each location, an \((s - 1, s)\) inventory control policy (one for one replenishment) is used;
5. there is no commonality, so a subcomponent may not be part of two different parent-components. This assumption is not critical, but some minor changes would be required and the problem size would increase;
6. replacement of a failed component by a functioning component takes zero time;
7. the repair lead time includes the time used for sending the failed component to the repair location and for diagnosing the failure cause;
8. the replenishment lead time for discarded components includes the time used for sending the failed component to the discard location;
9. all repair lead times, replenishment lead times, and move lead times (to move a functioning (repaired or newly purchased) component from a location to one of its child locations) are independent random variables;

10. repairs are always successful.

These assumptions are also made in the standard vari-metric type methods (see, e.g., Muckstadt, 2005; Sherbrooke, 2004).

3.2 Mathematical model

We introduce the notation in Section 3.2.1 and in Section 3.2.2, we give the mathematical model formulation.

3.2.1 Notation.

Let the set of all components be $C$, with $C_1 \subseteq C$ being the set of lrus. $\Gamma_c$ 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 $e_{\text{max}}$. The set $D$ consists of the possible decisions that can be made: $D = \{\text{discard, repair, move}\}$. 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 $D_{e_{\text{max}}} = \{\text{discard, repair}\}$.

Let the set $R$ consist of the resources. $\Omega_r$ is the set of tuples $(c, d)$ such that $(c, d) \in \Omega_r$ if and only if component $c \in C$ requires resource $r \in R$ in order to enable decision $d \in D$.

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, $X$ is the matrix of all variables $X_{c,e,d}$ and $S$ is the matrix of all variables $S_{c,e}$.

For each component $c \in C$, we define $\lambda_c$ as the total annual failure rate over all operating sites. We define three cost types. For component $c \in C$ at echelon level $e \in E$, $v_{c,e,d}$ 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 $\lambda_c$, we define $f_{r,e}$ to be the annual fixed costs to locate resource $r \in R$ at echelon level $e \in E$ and we define $h_{c,e}$ to be the annual costs of holding one spare of component $c \in C$ at each location at echelon level $e$.

3.2.2 Mathematical model formulation.

We define our model as follows:
minimize \sum_{c \in C} \sum_{e \in E} \sum_{d \in D} c_{c,e,d} \cdot \lambda_c \cdot X_{c,e,d} + \sum_{r \in R} \sum_{e \in E} f_{r,e} \cdot Y_{r,e} + \sum_{c \in C} \sum_{e \in E} h_{c,e} \cdot S_{c,e} \quad (1)

subject to:

\sum_{d \in D_e} X_{c,1,d} = 1, \forall c \in C_1 \quad (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_{\max} \quad (3)

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

availability(X, S) \geq \text{target availability} \quad (5)

\sum_{r \in R} \sum_{(c,d) \in \Omega_r} Y_{r,e} \leq \text{target availability} \quad (6)

X_{c,e,d}, Y_{r,e} \in \{0, 1\} \quad (7)

S_{c,e} \in \mathbb{N} \quad (8)

Constraints 2 to 5 are the ‘LORA constraints’ and define the same model as Basten et al. (2009b) 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 X and all spare parts decisions S, our model cannot be solved using an ILP solver and it cannot be decomposed per component. Instead, we use the VARI-METRIC method to calculate the availability. In order to have costs in the objective function that are comparable, we use the annual spare parts holding costs instead of the spare parts investment costs, which is common in VARI-METRIC. However, the difference is a constant factor.

3.3 Demarcation

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

- 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.

4 VARI-METRIC: the marginal analysis approach

In this section, we explain the logic of VARI-METRIC since our algorithm is inspired by it. A more detailed explanation of the exact steps to take in the VARI-METRIC algorithm can be found in Appendix A. We base our explanation mainly on Muckstadt (2005) and refer to his book or Sherbrooke (2004) for more information.

Given repair/discard decisions per component, the aim of VARI-METRIC is to stock spare parts in a repair network such that a target availability is achieved against the lowest possible spare parts investment costs. Assuming that replacement of a failed component with a
spare takes zero time, a capital good is unavailable only if no spare part is available at that operating site. In that case, a backorder arises. The expected number of backorders (EBO) is: \(\sum_{x=s+1}^{\infty} P(PI = x)\), with \(s\) being the number of spare parts that is initially located at the operating site and \(PI\) being the pipeline, so the number of spare parts in repair at the operating site or requested but not received yet from a location at the next higher echelon level.

The availability at the operating site is: \(\prod_{c \in C} (1 - \text{EBO}_c(s_c)/(N \cdot Z_c))^{Z_c}\), with \(N\) being the number of capital goods at the operating site and \(Z_c\) being the number of occurrences of LRU \(c\) in each capital good. As an approximation, the number of LRU backorders at operating sites equals the number of systems that are down waiting for a spare part. Therefore, maximizing the expected availability of the installed base is approximately equivalent to minimizing the expected backorders (EBO) of LRU s at operating sites (see, e.g., Sherbrooke, 2004, pp. 39–40).

**Vari-Metric** generates a convex trade-off curve of spares investment costs versus the expected LRU backorders at operating sites, a so-called **EBO-curve**.

We define an **LRU family** as an LRU including all its subcomponents at any indenture level. Since our objective is to minimize the summation of the EBOs for all LRUs at the operating sites, the overall spare parts stocking problem is separable in subproblems per LRU family, assuming that subcomponents are not shared between LRU families, i.e., there is no commonality.

Backorders at higher indenture levels in the LRU family or at higher echelon levels influence the availability of the installed base only in an indirect way, since they influence the throughput times of requests for LRUs at operating sites: the repair time may be increased due to subcomponents being backordered and supplies from the location at the next higher echelon level may be delayed due to backorders at that location. The **Vari-Metric** model provides expressions for the mean and variance of the number of items in the pipeline (spares that are requested, but not received yet, e.g., due to transportation times and backorders) and fits a negative binomial distribution to these first two moments. Generally, this is an accurate approximation for the true distribution of the pipeline.

For a single component, **Vari-Metric** works as follows. Starting at an initial stock (usually zero), we add spare parts to stock one-by-one. A spare part is added to stock at that location that yields the highest reduction of total expected backorders at the operating sites. If there are multiple echelon levels, we first add spare parts one-by-one at the highest echelon level. Then, for each number of spare parts at the highest echelon level, spare parts are added at the next lower echelon level, resulting in multiple EBO-curves. This is done recursively if there are more than two echelon levels. The key idea is now to determine the lower envelope of the resulting curves and remove all non-convex points so that one convex EBO-curve remains. The lower envelope of a set of functions \(f_i(x)\) is the function \(f(x)\) given by their pointwise minimum \(f(x) = \min_i f_i(x)\). Figure 3 gives an example of a number of EBO-curves. Figure 4 shows the same curves, including the lower envelope, and Figure 5 also shows the convexification of the lower envelope (points are removed such that a convex function remains). Notice that going from one point to the next on the convexified lower envelope, does not mean that one spare part is added. Each point represents a certain number of spare parts located throughout the repair network.
For a multi-indenture component, we construct an EBO-curve for each of the components at the highest indenture level as described above. We merge the results using marginal analysis. That is, we successively add points from the individual curves to the overall curve. We choose the first point from that curve that yields the highest ratio of decrease in expected backorders as experienced by the parent-component to increase in costs. Since these curves are convex, we may look ahead one step per curve only and the resulting overall curve will be convex too. Figure 6 gives an example of EBO-curves for two SRUs and Figure 7 shows the result of applying marginal analysis.

For each of the points on the curve for the child-components, we construct an EBO-curve for
the parent-component in the same way as described above, taking the backorders that result from the child-components into account. This results in a number of \textit{ebo}-curves, which means that we have to determine the lower envelope of these curves. If there are multiple indenture levels, we recursively merge results of child-components using marginal analysis and use the resulting curve to construct new curves for the parent-component.

Finally, if we have one convex \textit{ebo}-curve for each \textit{lru}-family, we use marginal analysis to get one convex trade-off curve of spares investment costs (resulting from components at all indenture levels) versus the expected \textit{lru} backorders at the operating sites. A point on this curve represents an \textit{optimal backorder level} for a certain investment level (except for the approximation errors that result from using the first two moments of the distribution of the pipeline only).

There are two further things to notice:

- the \textit{availability} is not necessarily optimal for a given investment level, since maximization of the availability is only approximated by minimization of the summation of \textit{lru} backorders;

- the \textit{ebo}-curve is not a continuous curve, but consists of a number of points only. As a result, the achieved availability may be higher than the target availability, which is called \textit{overshoot}. However, due to the number of points on the curve, this overshoot is usually small. The overshoot issue may be reduced further using the following heuristic. Account for the increase in availability upto the target availability only in the marginal analysis approach, since any additional increase in availability is ‘useless’, see also Kranenburg (2006, p.26). However, the resulting solutions do not represent optimal backorder levels for a given investment level, which is why we do not use this heuristic in our algorithm.

\section{Algorithm}

Section 5.1 describes the general idea of our algorithm. In Section 5.2, we show how this algorithm is used if all components in one \textit{lru} family together require at most one resource. Section 5.3 discusses the general case, in which any number of resources may be required.

\subsection{General idea}

In Section 4, we have shown how to determine spare parts stocks, given repair/discard decisions. A straightforward approach to solve the joint problem of \textit{lora} and spare parts stocking is to consider all combinations of repair/discard decisions for all components, stock spare parts for each combination, and determine the convexification of the lower envelope. Due to the number of possible combinations to consider, this approach is realistic for very small problem instances only. However, we can decompose the problem into subproblems, using an algorithm that is inspired by that of Alfredsson (1997). We use the same two types of decomposing the problem into subproblems, described in Section 2.3:
1. We decompose the problem into one subproblem per resource in the special case that we discuss in Section 5.2. In the general case, it is slightly more complicated, as we explain in Section 5.3. We further decompose the problem into one subproblem for each LRU as in standard METRIC type methods.

2. We fixate decision variables, thereby decomposing the problem into subproblems. For example, we fixate the location of a resource and the repair location of a component. We explain in Sections 5.2 and 5.3 when and how each type of decomposition should be used. Merging the results of the subproblems means performing marginal analysis for type 1 decompositions and finding the convexification of the lower envelope for type 2 decompositions.

5.2 Special case: at most one resource per LRU-tree

If all components in an LRU family together require only one resource, as in the example in Figure 8, we can decompose the problem as follows.

- Using a type 1 decomposition, decompose the problem into one subproblem for each resource. For each resource there are $2^{|E|}$ scenarios ($E$ is the set of echelon levels), resulting from either installing or not installing the resource at each of the echelon levels.

- Decompose (type 1 decomposition) each of the subproblems per resource into one subproblem for each LRU family, just as the standard METRIC type models do. Each LRU family that requires no resource at all, makes up such a subproblem too.

- Decompose (type 2 decomposition) each of the subproblems per LRU family into one subproblem for each repair/discard option by fixating for the LRU either a repair-location or choosing the discard option. Notice that due to the location of the resource (if any is required), some repair options may not be available.

- Decompose (type 1 decomposition) each of the subproblems per repair option for the LRU into one subproblem for each SRU. Notice that in the subproblem in which the LRU is discarded, nothing needs to be done for the SRUs.

- Decompose (type 2 decomposition) each of the subproblems per SRU into one subproblem for each repair/discard option for the SRU. Notice that due to the location of the resource (if any is required) and due to the repair location of the LRU, some repair options may not be available.

In Section 4, we explained that we have to construct an EBO-curve for each of the SRUs, merge these, and add spare LRU for each of the resulting points on the SRUs-curve. Here, it is somewhat more complicated. For each SRU, we construct one curve for each of the repair/- discard options. We find the convexification of the lower envelope of these curves, which leads to one convex EBO-curve for each of the SRUs. Merging the curves of the SRUs and solving the subproblem for the LRU is done as explained in Section 4. This results in one curve for each repair/discard option for the LRU. We find the convexification of the lower envelope over all options per LRU, merge the results per resource using marginal analysis, and finally, merge the total results using marginal analysis. The approach described above can be used recursively if there are multiple indenture levels in the product structure.
5.3 General case: multiple resources per LRU family

In general, one component may require multiple resources, or components in one LRU family may need various resources. In that case, resources cannot be considered independently of each other, but form one resource group. Consider the example product structure in Figure 9 and a three-echelon repair network. To repair LRU \( c_4 \), resources \( r_1 \) and \( r_2 \) are required, and to repair LRU \( c_5 \), resources \( r_2 \) and \( r_3 \) are required. As a result, the decision to install resource \( r_2 \) depends on the decision to install resource \( r_1 \) and vice versa. It also depends on the decision to install resource \( r_3 \) and vice versa. Therefore, resources \( r_1 \), \( r_2 \), and \( r_3 \) make up one resource group. Below, we give an exact definition of a resource group.

In Section 5.2, we showed that for each resource there are eight possible scenarios in a repair network with three echelon levels \( (2^E) \). If we consider all combinations of resource locations in our example, we get \( 8^3 = 512 \) scenarios, or \( 2^{|E|} \cdot |R^{group}| \), with \( |R^{group}| \) being the number of resources in the resource group (notice that in this specific example, locating resource \( r_2 \) at a certain echelon level, without locating at least one of the other resources at the same echelon level, is useless).

Since the number of scenarios explodes if the number of interacting resources increases, we decompose the problem into subproblems (type 2 decomposition). In our example, we observe that once we know where resource \( r_2 \) is located (which may be at multiple levels in the repair network), resources \( r_1 \) and \( r_3 \) can be treated independently of each other (but not independently of resource \( r_2 \)). Therefore, we make eight main scenarios in which one or more locations for resource \( r_2 \) are fixated. For each of these main scenarios, there are two subproblems: the first consists of resource \( r_1 \) and LRU families \( c_1 \) and \( c_4 \), the second consists of resource \( r_3 \) and LRU families \( c_5 \) and \( c_6 \). For each of these subproblems, there are eight scenarios, resulting in \( 8 \cdot (8 + 8) = 128 \) scenarios. Notice that in each of the 128 scenarios, we consider two LRU s as opposed to four LRU s for each of the 512 original scenarios.

In Section 5.2, we explained how to obtain the results for each of the subproblems (one resource and two LRU families). Using marginal analysis, we merge the results of the two subproblems to get a result for each main scenario. Finding the convexification of the lower envelope of the eight main scenarios leads to the overall EBO-curve for each resource-group.

Now the problem is how to decompose a general problem with shared resources into independent subproblems so that the total problem can be solved efficiently. To this end, we represent the interaction between the resources in a graph: a vertex represents a resource, and an edge between two vertices exists if there exists an LRU family that uses both resources. Figure 10 represents the graph for our example. Below, we first give a number of definitions, then we explain how we use the graph representation of the interaction between the resources. We assume a familiarity with the basics of graph theory; for further definitions, we refer to any book on general graph theory, for example, Diestel (2005) or Godsil and Royle (2001).

A graph \( G = (V, E) \) consists of vertices \( v \in V \) and edges between vertices \( (v, w) \in E \). A
graph $G' = (V', E')$, with $V' \subseteq V$ and $(v, w) \in E' \iff (v, w) \in E, v \in V', w \in V'$, is an induced subgraph of $G$. A graph is connected if any two of its vertices are linked by a path (the graph consisting of one vertex is connected as well). A maximal connected subgraph $G' = (V', E')$ is an induced subgraph of $G = (V, E)$, such that adding any more vertices $v \in V \setminus V'$ (and the required edges $(v, w) \in E \setminus E'$ to keep an induced subgraph) leads to a disconnected graph. A maximal connected subgraph is also called a connected component or just component; we will use the term graph component to avoid confusion. A depth first search can be used to identify the graph components in a graph (see, e.g., Hopcroft and Tarjan, 1973). A graph is complete if an edge exists between any two vertices: if for all vertices $v_1, v_2 \in V$ with $v_1 \neq v_2$ it holds that $(v_1, v_2) \in E$, then the graph $G = (V, E)$ is complete.

We now define a resource group as the set of all resources represented by the vertices in one graph component. We have already shown that the number of options for the resources in a resource group is $2^{|E|}$, where $|E|$ is the number of edges. We also know that fixating the location(s) for one resource leads to $2^{|E|}$ scenarios. Our goal is to decompose any resource group in such a way that we get the smallest number of scenarios. Notice that decomposition of the resource group is useful only if it is represented by a graph component that is not complete. We decompose the problem using a recursive approach on the graph representation of the resource group as follows.

1. Check whether the graph component, representing a resource group, is complete. If so, the number of scenarios is $2^{|E|}$. Otherwise, go to step 2.

2. For each vertex in the graph component, representing a resource, remove the vertex from the graph component. Using a depth-first search, find the graph components in the new subgraph. For each of the new graph components, go to step 1. The total number of scenarios is $2^{|E|}$ times the summation of the number of scenarios in each new graph components. Go to step 3.

3. Over all the vertices that can be removed, choose the vertex that leads to the smallest number of scenarios. This vertex represents the resource that should be fixated.

Since the number of resources is limited, this approach takes less than 0.1% of the total computation time of the algorithm in our experiments. Hence, it does not make sense to find a more efficient method for problem decomposition.

6 Test results

In this section, we answer three questions:

1. What problem size can our algorithm deal with and which model parameters influence the computation time that we achieve?

2. Using our algorithm as a benchmark, how is the performance of the iterative algorithm of Basten et al. (2009b)?

3. Which model parameters influence the performance of the iterative algorithm?

The problem instances that we use to answer these questions are generated using the generator that was presented by Basten et al. (2009b) and is described in detail in Appendix B. We summarize the key characteristics of the generator in Section 6.1. In Sections 6.2, 6.3, and 6.4, we answer the three questions, respectively. Then, in Section 6.5, we discuss the case study at Thales Nederland. If we use the iterative algorithm, we use the best variant found by Basten et al. (2009b). We solve problems instances on an Intel Core 2 at 2.13 GHz with 2 GB RAM, running Windows XP.

---

1 Notice that some scenarios may consist of many LRU’s, whereas others may consist of a few LRU’s only. One may want to incorporate this in the search for the best way of decomposing the resource group, but for sake of simplicity, we do not do that.

2 Do not remove a vertex that is connected to one other vertex only, since that does not help in decomposing the graph component.
Table 1: Cost difference and optimization time for each parameter setting

<table>
<thead>
<tr>
<th>Test set</th>
<th>Parameter</th>
<th>Setting</th>
<th>Run time (minutes)</th>
<th>Cost difference with iterative Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td># indenture levels</td>
<td>2</td>
<td>0.4</td>
<td>-0.21%</td>
<td>-4.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>4.3</td>
<td>-0.21%</td>
<td>-3.17%</td>
</tr>
<tr>
<td>1</td>
<td># LRU</td>
<td>50</td>
<td>3.8</td>
<td>-0.22%</td>
<td>-4.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0.9</td>
<td>-0.19%</td>
<td>-3.17%</td>
</tr>
<tr>
<td>1</td>
<td># echelon levels</td>
<td>2</td>
<td>0.1</td>
<td>-0.13%</td>
<td>-4.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>4.6</td>
<td>-0.28%</td>
<td>-3.24%</td>
</tr>
<tr>
<td></td>
<td>Move lead time</td>
<td>[0.5/52, 4/52]</td>
<td>2.2</td>
<td>-0.44%</td>
<td>-4.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2/52, 4/52]</td>
<td>2.5</td>
<td>0.02%</td>
<td>-3.03%</td>
</tr>
<tr>
<td>2</td>
<td>Demand per LRU</td>
<td>[0.01, 0.1]</td>
<td>8</td>
<td>-0.50%</td>
<td>-2.44%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.01, 0.25]</td>
<td>14</td>
<td>0.08%</td>
<td>-1.78%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.01, 0.5]</td>
<td>25</td>
<td>-1.35%</td>
<td>-2.53%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.01, 1]</td>
<td>53</td>
<td>-1.34%</td>
<td>-2.17%</td>
</tr>
<tr>
<td>3</td>
<td>Components per resource</td>
<td>[2, 3]</td>
<td>11</td>
<td>0.06%</td>
<td>-1.04%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2, 6]</td>
<td>30</td>
<td>-0.03%</td>
<td>-1.69%</td>
</tr>
<tr>
<td>3</td>
<td>% resources used by 1 component</td>
<td>0%</td>
<td>32</td>
<td>-0.06%</td>
<td>-1.69%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50%</td>
<td>10</td>
<td>0.09%</td>
<td>-1.05%</td>
</tr>
</tbody>
</table>

6.1 Generator

We divide the problem instances in three test sets. In each set, a number of parameters is varied (full factorial) in order to see the effect of that parameter on the results. The focus of each of the test sets is as follows:

1. The sizes of the product structures and network structures are varied, together with lead times and holding costs (1,280 problem instances).

2. Costs of components and resources and annual demands are varied (160 problem instances).

3. Component-resource relations are varied (80 problem instances).

An overview of the results for the various parameter settings can be found in Table 1. This table shows the parameters that significantly influence the results only. Appendix B contains a complete overview of the parameter settings.

6.2 Problem size and computation time

Our problem instances contain up to 700 components, three indenture levels, and three echelon levels, which is a size that is realistic in practice, e.g., at Thales Nederland. Due to the memory requirements of our algorithm, we are not able to solve problem instances that are much larger than the ones in our tests, using our test computer that has 2 GB RAM. The key reasons for high memory usage are the interacting resources and the EBO-curves that need to be constructed for multiple echelon and indenture levels. The maximum computation time over all problem instances is near six hours, in test set 2. Notice however, that a computation time of six hours need not be problematic for usage of our algorithm in practice since we deal with decisions at the tactical level that need to be made only occasionally.

Table 1 shows both the computation time and cost difference compared with the iterative algorithm for those parameters that have a significant impact on the computation time or the cost difference. Here, we are interested in the computation time only. In test set 1, the computation time is mainly determined by the number of indenture levels, LRU and echelon levels. Increasing the number of indenture levels leads to more components in the product
structure and increasing the number of echelon levels increases the number of possible repair/-
discard options, both resulting in a higher computation time. An increase in the number of LRU-s leads to a decrease of the computation time. The reason is that if the number of LRU-s increases, the probability of generating problem instances with one large resource group decreases, since the number of resources and resource-component relations does not change. Finding the EBO-
curve for a large resource group is computationally expensive, since many combinations of resource-locations need to be considered.

In test set 2, the computation time increases sharply with an increasing demand: if demand increases, the EBO increases and more spares need to be stocked. Due to calculating more points on the EBO-curves, the running time increases. In test set 3, the computation times are mainly determined by the number of components per resource, and the percentage of components that is used by one component only. An increase in the number of components per resource or a decrease in the percentage of components that is used by one component only leads to more resource-component relations, thus increasing the complexity of the problem and the average running time.

6.3 Benchmarking the iterative algorithm

On the generated problem instances in the three test sets, our algorithm gives an average cost reduction of 0.26% on average compared with the iterative method, with a maximum of almost 5%. This means that the iterative approach of Basten et al. (2009b) works pretty well in most cases. Still, the maximum cost deviation shows that there are problem instances on which the iterative algorithm does not find a solution that is close to optimal. On the other hand, we found some problem instances for which the integrated approach yields higher costs than the iterative approach. This is due to the overshoot problem that we explained in Section 4. Comparing the achieved availability, we see that the iterative approach finds solutions that are on average closer to the target availability of 95% than the integrated method: 95.07% and 95.20%, respectively (the maximum achieved availability is 96.12% and 96.24%, respectively).

Compared to the sequential approach, our integrated algorithm leads to a cost reduction of almost 3.40% on average, with a maximum of almost 37%. This is slightly more than Basten et al. (2009b) report using the iterative algorithm (3.17% on average and over 35% at maximum).

6.4 Influence of the parameter settings

Table 1 shows the cost difference between our integrated approach and the iterative approach of Basten et al. (2009b). In test set 1, the largest average cost reduction is achieved in problem instances with small move lead times. Basten et al. (2009b) explained the influence of the move lead time on the cost reduction that can be achieved using the iterative algorithm instead of the sequential approach: the iterative algorithm performs repairs (mainly of components for which no resources are required) at a higher echelon level, so that it can use pooling effects of the spare parts. However, the iterative approach does not use all pooling opportunities, which is why there is a cost difference with our approach, especially if the move lead times are low.

In general, if the problem size increases (number of indenture levels, number of LRU-s, and number of echelon levels) the maximum cost reduction decreases, which means that the performance of the iterative method in the worst case improves with an increasing problem size.

In test set 2, the cost difference is mainly determined by the demand per LRU. In test set 3, none of the parameter settings really influences the cost differences.

6.5 Thales case

We performed a case study at Thales Nederland, a manufacturer of naval sensors and naval command and control systems; the case is representative for the problems that Thales Nederland has to solve in general. Basten et al. (2009b) describe the case and the current way of working at Thales Nederland in more detail; we give the main characteristics of the case below.

Although the product structure is modelled at Thales using six indenture levels, there are effectively only three indenture levels. The other indenture levels consist of ‘structure parts’ that cannot be removed from the ship and small parts that will be discarded by default. For
the same reason, more components are used in the product, but only slightly more than 200 are modelled.

The repair network consists of twelve ships, attached to two intermediate depots, a central depot and the OEM. There are 54 resources, 34 of which are adapters. These adapters are used in concurrence with some expensive test equipment: the latter can be used to test a set of components, but distinct adapters need to be installed for each component that is actually tested.

It turns out that 50 resources form one resource group; four resources need to be fixated before the remaining graph components are complete. However, due to the adapters, many small subgraphs exist of only one resource. Still, it takes about two days to solve the case study, which makes it doubtful whether this method can be used in practice.

Figure 11 shows the costs that result from using each of the three methods to solve the case study. A cost reduction of 6.7% is achieved by our integrated method compared with the sequential method, which is worth millions over the life time of a sensor system. However Basten et al. (2008) found a cost reduction of 9.7% using the iterative algorithm in less than one and a half minute. The reason is the overshoot problem that we discussed before (Sections 4 and 6.3): the iterative method achieves an availability of 95.01%, whereas the integrated method achieves an availability of 95.37%. Figure 12 shows the availability that the integrated model achieves as a function of the total costs (all efficient points). We see that the availability is above 95.01% for the costs of 0.903 (the solution of the iterative method), which means that the iterative solution is indeed below the efficient frontier that is found by the integrated method. 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 (and the solution would be very close to optimal).

7 Conclusions and further research

In this paper, we presented an 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:

- Our method finds a solution that is optimal in the sense that the achieved backorder level cannot be realized against lower costs, and except for the approximation errors in the VARI-METRIC methods, which are known to be small. The reason is that we implicitly enumerate all possible LORA decisions in an efficient way.

- Therefore, we are also able to state that the quality of the iterative algorithm of Basten et al. (2009b) is quite good on average. It is, however, less robust than our integrated
method in the sense that the cost difference compared with the optimum can be considerable for specific problem instances. If the problem size increases (number of indenture levels, number of LRU's, and number of echelon levels), the maximal cost reduction decreases, which means that the performance of the iterative method improves in the worst case. Unfortunately, we are not able to distinguish classes of problem instances for which the iterative algorithm is guaranteed to find a solution that is close to optimal.

• Since our method requires much more time than the iterative method of Basten et al. (2009b) (e.g., on the case study at Thales Nederland we require about two days, opposed to about 10 minutes for the iterative algorithm) and the latter model leads to a smaller overshoot on average, we propose to use their method in practice during the development process and to use our method only at the major milestones in the development process to be sure that the results are correct.

Our recommendations for further research are as follows:

• Use our integrated method as a benchmark and develop heuristic methods that find a solution that is close to optimal, possibly by improving the iterative algorithm of Basten et al. (2009b).

• Improve the integrated method by using more of the structure of the problem. For example: some resources are so expensive that they will never be placed at all operating sites or at multiple echelon levels. The result may be an algorithm that is still guaranteed to be optimal (in the same sense as our algorithm is optimal) or a good heuristic.

• Extend our algorithm to cover non-symmetrical networks. This is straightforward from a modelling perspective, but it will lead to an increase of the computation time and memory usage.

• Improve the efficiency of our algorithm, especially if our algorithm would be extended to cover non-symmetrical networks.

Acknowledgement

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.

A VARI-METRIC

In this appendix, we assume that the probability that a component is repaired at a certain location is always either 0 or 1 ($r \in \{0,1\}$ in VARI-METRIC terms), which fits our joint model.
of LORA and spare parts stocking and makes the explanation simpler. If the aforementioned probability is either 0 or 1, demand for some components may be zero at some locations and stocking this component at that location makes no sense: if a component is replaced at location $l^e$ and repaired or discarded at location $l^r$, then there will be one path in the repair network between these two locations and stocking the component makes sense only at any location on that path, including locations $l^e$ and $l^r$. For example, if in a three-echelon repair network an LRU is repaired at the intermediate depots, it may only be stocked at the operating sites or the intermediate depots. If the LRU consists of one SRU that is discarded, that SRU may be stocked at the intermediate depots and the central depot. We call these echelon levels the relevant echelon levels. Recall that the marginal analysis approach of VARI-METRIC leads to a trade-off curve of spare parts investment costs versus backorders, a so-called EBO-curve.

Step 1: Construct an EBO-curve for each highest indenture level component.\(^3\)

1a: Add components to stock one-by-one at the highest relevant echelon level. The component is added to stock at that location that yields the highest reduction of total expected backorders at this echelon level. Go to step 1b.

1b: The backorders on the EBO-curve(s) that we constructed increase the waiting time for either this component at the next lower echelon level or for its parent-component (if the parent-component is repaired at the current echelon level). In the latter case, go to step 1d; in the former case, go to step 1c.

1c: For each point on the EBO-curve that we already have, construct an EBO-curve by adding components to stock one-by-one at the next lower echelon level. The component is added to stock at that location that yields the highest reduction of total expected backorders at that echelon level. Go to step 1b.

1d: At this point, there may be either one or multiple EBO-curve(s). If there are multiple EBO-curves, we determine the lower envelope of the resulting curves and remove all non-convex points so that one convex EBO-curve results. We now have one convex EBO-curve representing optimal backorder levels of one component as experienced by its parent-component versus investment costs. Go to step 2.

Step 2: Merge curves for all components that are a subcomponent of the same parent-component. If the components are LRU-s, the ‘parent-component’ has the meaning of ‘total capital good’.

2a: We merge the curves using marginal analysis. That is, we successively add points from the individual EBO-curves to the overall EBO-curve. We choose the first point from that curve that yields the highest ratio of decrease in total expected backorders experienced by the parent-component to increase in costs. Since these curves are convex, we need to look ahead one step per curve only, and the resulting overall curve will be convex too. If the parent-component is the total system, so we are merging the results from LRU-s, we are done. Otherwise, go to step 3.

Step 3: Construct an EBO-curve for the parent-component.

3a: For each point on the EBO-curve that we have for the total of all subcomponents, add components to stock one-by-one at the highest relevant echelon level for the (parent-)component. The component is added to stock at that location that yields the highest reduction of total expected backorders at that echelon level. Go to step 1b.

B Problem instances generator

We explain how we generate the problem instances that we use in our experiments. The problem instances are divided into three test sets, each with its own focus. We vary in each test set:

1. sizes of the product structures and network structures, lead times, and holding costs;
2. annual demands and costs of components and resources;
3. component-resource relations.

\(^3\)We consider components that have subcomponents, but are discarded, as the highest indenture level components, since their subcomponents will not be stocked.
For each parameter that we use to generate these instances, we use the default setting in the text. Some values are set to a certain value, others are drawn from a given range. These drawn values are the same for all settings of the other parameters. Table 2 (3) summarizes for all parameters that do not vary (do vary) their default setting, the alternative setting that we use, and the test set in which that alternative setting is used. We use a full factorial design and we generate 10 problem instances for each combination of parameters to decrease the risk of basing conclusions on one odd problem instance. As a result, test set 1, 2, and 3 consist of 1,280, 160, and 80 problem instances, respectively. Since the default value for each parameter is used in each test set, each test set contains the same ten ‘default’ problem instances.

We use a three-echelon repair network that is completely symmetrical in the cost factors, the demand rates, and the throughput times. It consists of a central depot, two intermediate depots, and ten operating sites. The three-indenture product structure consists of 50 LRU, 100 SRU, and 200 parts (so two subcomponents per component on average). Since each subcomponent is randomly assigned to one of the components, the number of subcomponents per component may differ for the various components. The annual demand for a component is equal to the annual demand of its subcomponents. We achieve this by drawing the annual demand of each part from a uniform distribution on the interval \([0.01/(\text{#subcomp. per comp.})^2 – 0.25/(\text{#subcomp. per comp.})^2]\), 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 \(7/(100,000 – 1,000)\). 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 100,000, but we draw a new price for these components to avoid odd problem instances. Using the calculated net 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 \([0.25 – 0.75]\).
- Next, we recursively add the costs of each subcomponent to its parent to get the gross component price for the parent.
- The discard costs as a fraction of the gross component price are drawn from a uniform distribution on the interval \([0.75 – 1.25]\). These costs include the costs of purchasing a new component and the disposal costs or residual value of the defective component.
- The move costs as a fraction of the gross component price are 1%.
- The annual costs of holding one spare part of a component are 20% of the gross component price.

For each component, we draw a repair time from a uniform distribution on the interval \([2/52 – 4/52]\). The discard time, the time it takes to order a component and receive it at the central depot, is drawn from a uniform distribution on the interval \([1/10 – 1/2]\). Both the discard and the repair times vary over the components, but are the same at all echelon levels. The replenishment lead time from one echelon to the next is drawn from a uniform distribution on the interval \([2/52 – 4/52]\). This value varies over the echelon levels, but is the same for all components.

There are ten resources and their annual costs are drawn from a shifted exponential distribution with shift factor 10,000 and rate parameter \(7/(100,000 – 10,000)\). We draw a new price for the resources that get a price higher than 100,000 to avoid odd problem instances. 50% of the resources will be used by one component only, the other 50% will be used by 2 to 6 components. We distinguish 4 ‘component types’, for example electronic versus mechanical components. Each resource and each LRU family is randomly assigned to one of the component types. The result is that resources of one component type do not interact with resources of another component type.
Table 2: Fixed values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Test set</th>
<th>Additional value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td># Echelon levels</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td># Central depots</td>
<td>1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td># Intermediate depots</td>
<td>2</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td># Operating sites</td>
<td>10</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td># Indenture levels</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td># LRU</td>
<td>50</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td># Subcomponents per parent component</td>
<td>2</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td># Resources</td>
<td>10</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td># Component types</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>% Resources used by 1 component</td>
<td>50%</td>
<td>3</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 3: Values that vary over a range

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default range</th>
<th>Test set</th>
<th>Additional value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual demand of LRU</td>
<td>0.01 – 0.25</td>
<td>2</td>
<td>0.01 – 0.1 &amp; 0.01 – 0.5 &amp; 0.01 – 1</td>
</tr>
<tr>
<td>Net cost of component</td>
<td>1,000 – 10,000</td>
<td>2</td>
<td>1,000 – 100,000</td>
</tr>
<tr>
<td>Discard costs</td>
<td>75% – 125%</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Repair costs</td>
<td>25% – 75%</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Move costs</td>
<td>1% – 1%</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Annual holding costs</td>
<td>20% – 20%</td>
<td>1</td>
<td>20% – 40%</td>
</tr>
<tr>
<td>Annual cost of resource</td>
<td>10,000 – 100,000</td>
<td>2</td>
<td>10,000 – 500,000</td>
</tr>
<tr>
<td>Discard time (in years)</td>
<td>1/10 – 1/2</td>
<td>1</td>
<td>1/4 – 1/2</td>
</tr>
<tr>
<td>Repair time (in years)</td>
<td>0.5/52 – 4/52</td>
<td>1</td>
<td>2/52 – 4/52</td>
</tr>
<tr>
<td>Move time (in years)</td>
<td>0.5/52 – 4/52</td>
<td>1</td>
<td>2/52 – 4/52</td>
</tr>
<tr>
<td># Components per resource</td>
<td>2 – 6</td>
<td>3</td>
<td>2 – 3</td>
</tr>
</tbody>
</table>

References


