



Innovative Applications of O.R.

## Defining line replaceable units

J. E. Parada Puig<sup>a,b,\*</sup>, R. J. I. Basten<sup>c</sup><sup>a</sup> Faculty of Engineering Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands<sup>b</sup> Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida, 5101, Venezuela<sup>c</sup> School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

### ARTICLE INFO

#### Article history:

Received 28 August 2014

Accepted 22 May 2015

Available online 3 June 2015

#### Keywords:

Maintenance

Replacement

Integer programming

Line replaceable unit definition

### ABSTRACT

Defective capital assets may be quickly restored to their operational condition by replacing the item that has failed. The item that is replaced is called the Line Replaceable Unit (LRU), and the so-called LRU definition problem is the problem of deciding on which item to replace upon each type of failure: when a replacement action is required in the field, service engineers can either replace the failed item itself or replace a parent assembly that holds the failed item. One option may be fast but expensive, while the other may take longer but against lower cost. We consider a maintenance organization that services a fleet of assets, so that unavailability due to maintenance downtime may be compensated by acquiring additional standby assets. The objective of the LRU-definition problem is to minimize the total cost of item replacement and the investment in additional assets, given a constraint on the availability of the fleet of assets. We link this problem to the literature. We also present two cases to show how the problem is treated in practice. We next model the problem as a mixed integer linear programming formulation, and we use a numerical experiment to illustrate the model, and the potential cost reductions that using such a model may lead to.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

### 1. Introduction

To maintain capital assets, a typical maintenance organization repairs them by replacing failed items (repair-by-replacement). A physical item that is replaced is called a *line replaceable unit* (LRU; see, e.g., DoD, 1996). The LRU definition problem is a maintenance policy decision that should be considered as a part of strategic or tactical maintenance planning: the exchange of LRUs produces downtime, and therefore the selection of items that should be defined as LRUs is a critical decision. Downtime can be compensated for with spare assets, and this means that the LRU decision should be considered from the outset of a capital asset acquisition program.

Traditionally, non-economic criteria are used to define LRUs. For example: Is it possible to know (test) that the item requires maintenance? Can the failed item be disassembled, and a spare reassembled to the asset without destruction or damage to other parts? Are there special adjustment and calibration needs? These technical criteria help engineers fit the LRU definition to existing practices and available resources of the maintenance organization. While these non-economic criteria are of key importance, inclusion of economic criteria can lead to a more cost effective LRU definition. The aim of this paper is to take a step in that direction.

We first link the problem to the scientific literature. Three relevant literature streams are reviewed: (i) maintenance task analysis, (ii) maintenance optimization, and (iii) level of repair analysis. The setup of this review is based on the Logistics Support Analysis framework (see, e.g., Jones, 2006). We find that the LRU decision is implicit in existing models for maintenance planning, and thus has not received the attention that it requires.

We next show how the problem is treated in practice by gathering insights from two organizations: a system developer, Thales Nederland BV, and a maintenance service provider, NedTrain BV. We show how LRU decisions are made at these organizations, giving insights about when they make the decision, who makes the decision, and what criteria are used. Also here, we find that the LRU definition decision is often made implicitly.

We propose to model the LRU definition problem explicitly. Using insights from the literature and from practice we come up with a *mixed integer linear programming* (MILP) formulation to find the optimal LRU definition. We perform a numerical experiment using typical problem sizes and parameters as they appear at NedTrain. Our theoretical contribution is as follows:

1. We link the problem to multi-component maintenance optimization and frame it in the literature as a decision that should be made after maintenance task analysis, and before level of repair;

\* Corresponding author. Tel.: +3153 489 2520; fax: +3153 4893631.

E-mail address: [j.e.paradapuig@utwente.nl](mailto:j.e.paradapuig@utwente.nl), [jorgep@ula.ve](mailto:jorgep@ula.ve) (J.E. Parada Puig).

2. We improve the LRU definition decision that is traditionally technical, by explicitly modeling the trade-off between downtime and cost, including replacement lead time, spare assets and the cost of replacement;
3. In multi-component maintenance optimization, the interactions between components are modeled. We explicitly incorporate one type of interaction called structural dependence, in which defining *what* to replace depends on the assembly structure of the capital asset.

From a practical point of view, we contribute by examining the cost savings that could be achieved compared to ad-hoc decisions made by experts. We do this in an extensive numerical experiment. We thus show that it is important to make the LRU definition decision explicitly in practice, and we give a model that can be used to do this.

The remainder of this paper is structured as follows. Section 2 presents the relevant literature and frames the LRU definition problem in the literature. Section 3 then shows two example cases from practice. Section 4 presents the LRU model notation, assumptions, and the mathematical formulation. Appendix B shows that the resulting LRU definition problem is NP-hard. Section 5 presents the numerical experiment. Finally, Section 6 discusses the conclusions and perspectives for future research.

## 2. Literature background

We use the framework of *logistics support analysis* (LSA) to structure our review of the literature. The LSA framework is shown in Fig. 1. It structures the decisions needed to produce the maintenance program for an asset, including the required (amounts of) resources. This enables us to position the LRU definition problem in the literature.

We first explain the LSA framework in Section 2.1. We then focus on three topics in detail; on maintenance task analysis in Section 2.2, on maintenance optimization, which covers the LRU definition problem, in Section 2.3, and on level of repair analysis in Section 2.4.

### 2.1. Logistic support analysis

Jones (2006) and Blanchard and Fabrycky (2011) provide good overviews of the LSA framework. It begins with the analysis of possible failure events. *Reliability predictions* are made for the failure of asset components. Next, maintenance significant items (and their failure effects and criticalities) are identified with the help of *fault-tree analysis* (FTA) and *failure modes, effects and criticality analysis* (FMECA). The analysis results are combined in the *reliability centered maintenance analysis* (RCMA) to establish the set of feasible maintenance policies for the capital asset, e.g., time based maintenance or run-to-failure (see, e.g., Moubrey, 1997; Tinga, 2010). At this point in the LSA framework, engineers have thus determined which items may fail, how often that is expected to happen, what effect and criticality such failure may have, and what preventive measures (if any) to take.

The next three analyses, *Maintenance task analysis* (MTA), *maintenance optimization* and *level of repair analysis* (LORA), are discussed in detail in the next three sections. MTA helps to identify and quantify the required maintenance resources, such as manpower or support equipment. Maintenance optimization models are mainly used to determine the optimal preventive maintenance intervals and task clustering. LORA supports repair or discard decisions, and determines where in the repair network to carry out these activities.

*Sparing analysis*, which follows after LORA, helps determine the spare parts package (see, e.g., Basten & van Houtum, 2014; Muckstadt, 2005; Sherbrooke, 2004, for an overview of the literature on spare parts inventory control models). *Life cycle cost* (LCC) is determined next. Finally, *value engineering* (VE) highlights asset functions that add cost but do not add significant value and feedback is given to design.

### 2.2. Maintenance task analysis

Maintenance task analysis is the detailed, step-by-step analysis of a maintenance task to determine how it should be performed, who will be required to perform it, and what physical resources are needed to complete it. Most maintenance tasks involve manual disassembly and (re)assembly operations. To find the best task procedure for maintenance, engineers use human factors analysis, path/motion planning and assembly/disassembly sequencing.

Human factors analysis helps to assess the effort to access the maintenance point and the risks involved, given a proceduralized task (see, e.g., Dhillon & Liu, 2006). Together with human factors, path and motion planning helps to reveal the best way for a service engineer to reach and route a part into or out of an assembly. Next, optimal sequencing helps to establish the optimal order of assembly and disassembly (see, e.g., Lambert, 2003).

Once the task procedure is established, maintainability analysis is used to estimate (or measure) the required time and resources. The literature on maintainability analysis has mostly concentrated on estimating the (mean) time to repair, using either statistical methods or expert-based assessment (see, e.g., Barabadi, Barabady, & Markeset, 2011; Moreu De Leon, González-Prida Díaz, Barberá Martínez, & Crespo Márquez, 2012). A quantification of both resource demand and task time are very useful for decision making. The data will be used as input of maintenance optimization models. We will need the results from MTA for solving LRU definition problem.

### 2.3. Multi-component maintenance optimization

Most literature on maintenance optimization focuses on defining the best policy for *when* to replace a particular item. However, for multi-component assets it is important to define not only when, but also *what* to replace. This derives from the fact that in capital assets with many items, interaction between items influences the maintenance action that should be chosen. Nicolai and Dekker (2008) review the literature on multi-component maintenance optimization

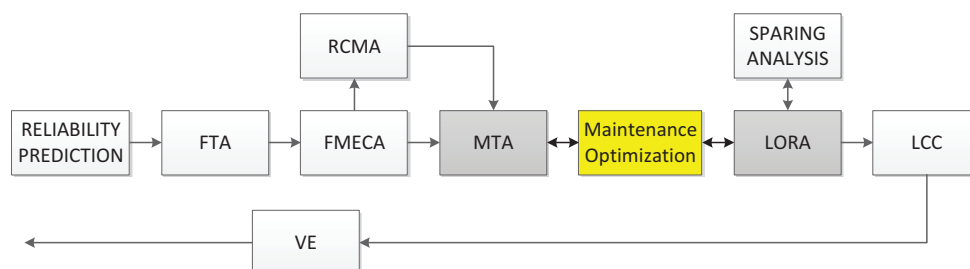


Fig. 1. The logistics support analysis framework (Adapted from Jones, 2006, page 11.23, Figure 11–16).

and they distinguish between economic, stochastic and structural dependence as a result of interaction between items.

Economic dependence exists when costs decrease or increase by grouping maintenance tasks. Stochastic dependence exists when items present failure interactions, i.e., states of items can affect the states of other items and their failure rate. Finally, structural dependence exists when items have to be replaced, or at least dismantled, before failed components can be replaced or repaired.

The term structural dependence was coined by Thomas (1986), who was one of the first to review the problems in multi-component maintenance optimization with structural dependence, i.e., where, “...the question is whether one should replace the whole car, the engine or just the piston rings when the piston rings need to be replaced”. With structural dependence, the disassembly precedence relations are important, which follow from MTA. We see the LRU definition problem as a multi-item maintenance optimization problem with structural dependence.

This problem has been a recurring problem in the literature on maintenance optimization for several decades, starting with the seminal paper by Sasieni (1956) (see e.g. Nicolai & Dekker, 2008; Marais, Rivas, Tetzloff, & Crossley, 2013). Moreover, while typically the focus has been on the decision about the optimal maintenance interval, i.e., when to replace, the key decision for the LRU definition problem is at what level within the indenture structure of a physical asset to define the LRU, i.e., the decision on what to replace.

2.4. Level of repair analysis

The level of repair analysis problem is a cost minimization problem that involves two decisions. Firstly, determining whether a maintenance significant item should be repaired or discarded upon failure. Secondly, determining where to allocate this repair/discard task in the repair network. The LORA problem has been addressed in the literature as an MILP model by several authors (see, e.g., Barros & Riley, 2001; Basten, van der Heijden, & Schutten, 2011). Recently, LORA models have been extended to consider availability by incorporating the amount of spare parts to stock (the sparing decision) (see, e.g., Basten, van der Heijden, & Schutten, 2012). We find that the literature on LORA takes the LRU decision as a given.

To the best of our knowledge, the importance of optimally determining line replaceable units was brought to attention only by Jensen (1975), in the context of level of repair analysis. Jensen states that the LRU definition is implicit in his LORA formulation. However, one of the inputs of his model is the explicit definition of which item is replaced directly from the asset, and which items (if any), are used to repair the failed LRU. Therefore, the LRU definition decision must be made before his LORA model can be used.

3. Defining LRUs in practice

We show insights about how organizations make the LRU definition decision in practice based on two exploratory cases. Interviews were conducted following the methodology in Schotborgh, McMahon, and Houten (2012). This means that we begin by asking experts to compare performance of different LRUs, and next focus on what performances they consider, e.g. time to replace, resources needed, task difficulty. We thus determine (i) when LRU definition decisions are made, (ii) who makes them, and (iii) what criteria are considered.

In Sections 3.1 and 3.2, we present our findings at a high-tech system developer and a maintenance service provider, respectively. For each case, we give a short company overview, followed by a description of the LRU definition process. Next, we discuss the criteria used by experts, and we give individual case conclusions. Finally, in Section 3.3 we draw more general conclusions from practice.

3.1. High-tech systems developer: Thales Nederland B.V.

Thales Nederland B.V. (Thales) is the largest defence company in the Netherlands. Thales designs and manufactures naval command and control, sensor, and communications systems. As a high tech systems developer, Thales is involved in maintenance by providing training, supplying service parts (LRUs), overhaul, upgrades, and modifications to its clients, and by performing repairs according to support contracts.

At Thales, design tasks are partitioned according to the type of technology used, and engineers are grouped into Technical Units with expertise in each type of technology, e.g., processing or microwaves. Fig. 2 shows how the LRU definition process is organized at Thales, according to the V-model of systems engineering. Through the

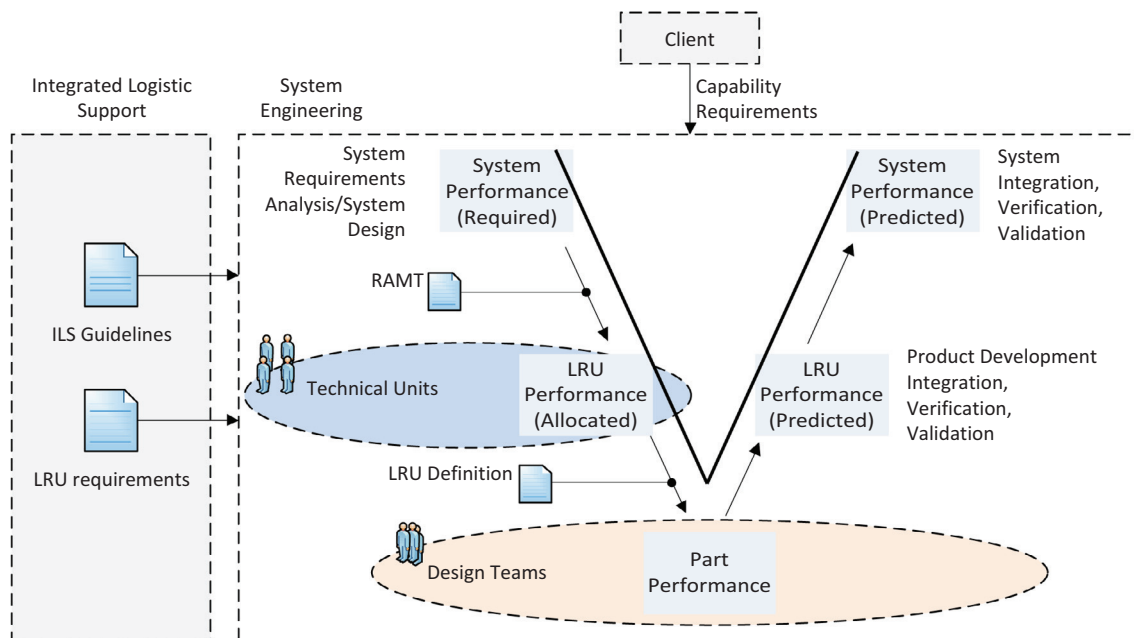


Fig. 2. LRU decisions in the workflow of Thales B.V.: from Functional Requirements (FR) to Design Solution (DS). Allocation of performance —Reliability, Availability, Maintainability, Testability (RAMT)— from system level to LRU level.

process of system requirements analysis, engineers translate the client's capability requirements into system performance. Maintenance performance is allocated at system level in the form of *Reliability, Availability, Maintainability, Testability* (RAMT) requirements.

Next, each Technical Unit must allocate the system level RAMT requirements to individual LRUs, thereby making the LRU definition decision. They use for this the (i) LRU requirements and the (ii) Integrated Logistic Support (ILS) Guidelines that have been developed by the ILS department. The Technical Unit communicates the LRU definition decisions to individual design teams. Design Teams will then develop the LRUs (parts) complying with the requirements and the logistics guidelines. For completeness, we include a summary of the LRU requirements in [Appendix D](#).

The ILS Guidelines are used to give an overview of supportability aspects to be considered by designers, as well as performance targets for design. Their goal is to improve system supportability and lower life cycle cost. Improving LRU definitions at Thales often involves redesign, and this is a considerable effort. Thales outsources manufacturing of some of its components, and this means that redesign has to involve the suppliers. Engineers can use the guidelines to trade-off manufacturability with supportability, i.e., a trade-off between manufacturing and operating costs.

The LRU requirements describe two types of criteria used by design teams when developing LRUs. Firstly, an LRU has to satisfy four *mandatory criteria*. If this is impossible, then the next higher assembly/equipment will be defined to be the LRU. Otherwise, the maintenance concept is changed to direct repair of the hardware item while installed in the asset, i.e., no repair by replacement and no definition of LRU.

Secondly, there are a number of *preferred criteria* for selection of LRUs. For example, there exist requirements for the maximum weight and dimensions. These allow the LRU to be handled by one service engineer only. If the design team has a good reason not to fulfill the requirement, then the preferred criteria may be waived. However, there is a constraint on the total number of waivers. Waivers are controlled during design review.

In summary, we have found that LRU definitions at Thales are made during the early stages of design. Engineers within technical units make the LRU definition decisions, and criteria for defining LRUs are based on compliance to standard LRU requirements. These requirements do not explicitly mention economic trade-offs.

### 3.2. Rolling stock maintenance service provider: NedTrain B.V.

*NedTrain B.V.* (NedTrain) is a full maintenance, repair and overhaul (MRO) service provider for trains. It is a subsidiary of NS, the main passenger railway operator in the Netherlands. NedTrain provides 24/7 service involving regular maintenance, repair, overhaul, modernization and life extension. Four types of facilities provide the required services, spanning from first line service to component repair and overhaul.

For NedTrain, the supportability of new trains depends on the level in the indenture structure at which LRUs are defined. The LRU level is initially defined during acquisitions of new trains, resulting in a preliminary list of LRUs. The preliminary list of LRUs may change throughout a train's life cycle when new knowledge about item performance becomes available. The actual LRU decision is made at the operational level. In practice, the LRU level is changed ad-hoc by service engineers during repair.

[Fig. 3](#) shows how the LRU definition process takes place during acquisition of new trains to produce the preliminary list of LRUs. A similar process is followed to change the list later in the life cycle. The LRU level is initially defined by the supplier during acquisitions, and it is communicated to NedTrain as a recommended spare parts assortment. For each of the suggested LRUs, NedTrain must decide to accept or revise this LRU level.

If NedTrain decides to revise the suggested list of LRUs, then experts will try to find an LRU level that gives a better fit to existing resources and to the maintenance concept. The whole process involves frequent communication with the suppliers. A team of experts begins the LRU research. This is typically done by looking deeper in the indenture structure to try to find smaller LRU candidate than the one proposed by the supplier. Next, they determine whether or not there are parts in this new level that *can* be exchanged. This is the most important technical aspect considered for defining LRUs. If there is no feasible disassembly sequence to remove the LRU candidate item directly, then the item is non-LRU. In this case the team shifts focus to the parent (assembly) item and repeats the analysis.

For some LRU candidates, direct replacement is possible only when the proper tools or equipment are available. If resources facilitate replacement and they are available, then the item is selected as an LRU. Some of the aspects considered for this decision include task frequency, skill level and effort required, for example. It is a rule of thumb that approximately 85 percent of the parts can be replaced in any maintenance workshop of NedTrain. However, the problem of limited availability of skills and manpower was noted by several experts.

If the required resources are not readily available in the workshop, then there is a cost trade-off. The expert team filters out those items that require expensive tools or equipment. If the cost of installing the additional resource is unacceptable, the expert team will begin negotiations with the supplier to outsource replacement. If an agreement is reached with the supplier, the item will be defined as LRU. In this case a service contract is signed and the supplier will perform direct replacement on the train. Otherwise, the item is defined as non-LRU.

The preliminary list of LRUs lends the operational LRU decision to service engineers in the workshops. Once the trains are fielded, there is a lot of freedom for service engineers at NedTrain to decide how a particular failure is repaired. LRU definitions are changed ad-hoc, i.e., the decision of what component is replaced. Often, if a service engineer decides to correct the failure by replacing an O-ring, (s)he can. If otherwise (s)he decides to replace the entire (assembled) unit, then that may also be allowed.

Service engineers find it is easier to replace a small part instead of a large assembly. This is because typically less interfaces have to be taken apart by replacing smaller items. Also, large and heavy components need more handling effort. However, some experts suggest that while replacing a small component is cheap, the train often remains waiting longer. The behavior of service engineers may come in conflict with this point of view. Our conclusion is that the LRU definitions should be a strategic/tactical decision. This is also the position of management.

Summarizing, we have found that LRU definitions decisions at NedTrain are made during acquisition and during the operating life of passenger trains. The goal during acquisitions is to analyze the preliminary list of LRUs of the supplier to create an initial spare parts assortment that better fits the maintenance concept of NedTrain. Therefore, the objective is not the specific choice amongst replacement alternatives. This LRU definitions are made by an expert team of engineers, in agreement with the suppliers, to fit the resources of NedTrain. In practice, during the operating life of the train the service engineers may make the replacement decision ad-hoc, and the LRU level within the indenture structure may change.

### 3.3. Case conclusions

Our research shows that LRU definitions require considerable efforts, that costs are not always incorporated and that supplier involvement is required. At NedTrain, performing the LRU definition decision for a new train series requires one year of an expert team's efforts. In practice, RAMS/RAMT influence LRU definitions, and LCC calculations have a limited influence in the decision. Both for Thales



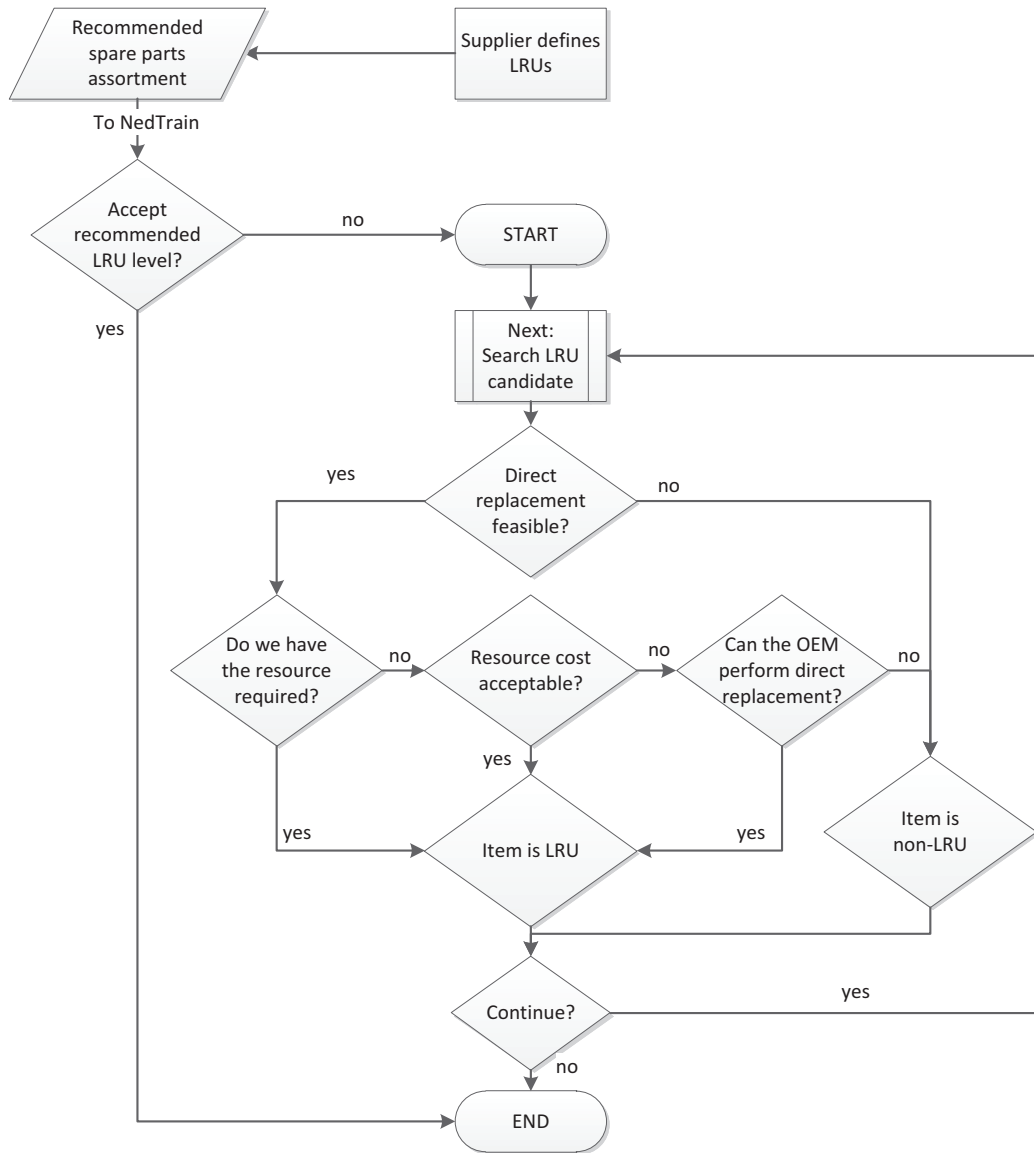


Fig. 3. First LRU decisions at NedTrain B.V.

and for NedTrain, increasing collaboration with suppliers has led to improved LRU definitions, e.g., higher availability and lower LCC.

To help Thales and NedTrain, we propose to use model-based decision support. We use an optimization approach in Section 4 of this paper which will probably be more useful for companies like NedTrain, though we expect it will also give insights that help designers at Thales. NedTrain can benefit from reduced downtime and increased standardization of repair. At Thales, helping designers make logistic trade-offs is required to support the system engineering efforts. This could be subject of future research.

4. Model

In Section 4.1, we explain the relevant notation and assumptions, and in Section 4.2, we present a mixed integer linear programming formulation for the LRU definition problem. Appendix B shows that the resulting LRU definition problem is NP-hard.

4.1. Notation and assumptions

Consider a capital asset with several indenture levels. Such a multi-indenture structure is a rooted ordered tree. Fig. 4 shows

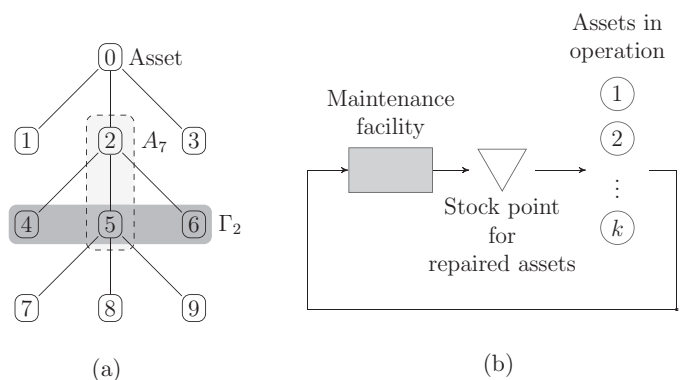


Fig. 4. (a) Example indenture structure (e.g.,  $I = \{1, \dots, 9\}$ ,  $\Gamma_2 = \{4, 5, 6\}$ , and  $A_7 = A_9 = \{2, 5\}$ ), and (b) a maintenance system for capital assets.

an example three-indenture asset with the notation that we use. Let  $I$  be the set of all maintenance significant items, and  $|I| \in \mathbb{N}$ , with  $|I|$  denoting the cardinality of  $I$ . For convenience we denote  $I = \{1, 2, \dots, |I|\}$ . Let  $\Gamma_i$  be the subset of items that are direct descendants of item  $i \in I$ . Notice that  $\Gamma_i = \emptyset$  if and only if item  $i \in I$  is a leaf

(i.e., has no descendants). We denote by the set  $A_i$  the set of all ascendants of item  $i \in I$ , i.e., all assemblies that contain, at some indenture level, item  $i$ .

Consider the maintenance system in Fig. 4. For its primary process, an operator requires  $k$  assets to be operational, i.e., the target number of available assets is  $k$ . Each of the  $k$  assets will remain in operation until a corrective maintenance action or preventive maintenance is required. When a maintenance event is required, the asset is removed from service and taken to a maintenance facility. After maintenance, the asset is sent to an inventory stock point.

We make some assumptions in order to keep our model simple, allowing us to focus on the insights that we can get. We assume that there is ample repair capacity (i.e., uncapacitated resources) and that item replacement times are deterministic with lead time  $r_i > 0$  for LRU  $i \in I$ . The item replacement time is the total time required for failure detection, isolation, repair and checkout/calibration. We assume that the asset behaves as a series of critical items: if one fails, then the asset fails. Finally, we assume that failures happen individually.

Let  $m_i$  be the individual failure rate (yearly number of failures) of item  $i \in I$ . This means that if  $m_i > 0$ , then failures occur in item  $i$  that are not due to failures in any of its descendants. Without loss of generality it holds that  $m_i > 0$  for all leaf items, i.e., items  $i \in I$  with  $\Gamma_i = \emptyset$ . We assume  $m_i$  over  $k$  operational assets. There may be moments when there are less than  $k$  assets operational, but given a high service level, this does not happen often and the number of operational assets will not be far below  $k$  (for a more extensive discussion of why such a constant failure rate is realistic to assume, see Sherbrooke, 2004, p.24, or Basten & van Houtum, 2014, p.40). Let  $\lambda_i$  be the cumulative failure rate of item  $i \in I$ .  $\lambda_i$  is the sum of the cumulative failure rates of those child items  $j \in \Gamma_i$  that are not defined as LRU.  $\lambda_i$  is thus an auxiliary variable and we show our recursive approach for calculating  $\lambda_i$  in Section 4.2. By definition,  $\lambda_i \geq m_i$ , for all  $i \in I$ , and  $\lambda_i = m_i$  if item  $i$  is a leaf item, i.e., if  $\Gamma_i = \emptyset$ .

We define the following decision variables:

- $X_i = \begin{cases} 1, & \text{if component } i \in I \text{ is defined as LRU,} \\ 0, & \text{otherwise;} \end{cases}$
- $N \in \mathbb{N}$  is the total number of assets to acquire.

The cost function contains two cost factors. Firstly, the annual holding cost,  $c_0$ , of the assets (this includes interest, depreciation, etc.). Secondly, the cost of replacement of an LRU  $i \in I$ ,  $c_i$ . Replacement costs include labor and material costs.

#### 4.2. Mixed integer linear programming formulation

Our objective is to determine which items out of the total set of items  $I$  to define as LRU, plus the total number of assets  $N$  to acquire, such that the total costs are minimized while the availability constraint is met. We state the LRU-definition problem as a mixed integer linear program below. The linearization of the objective function and Constraints 2–4 can be found in Appendix A.

$$\text{minimize } c_0 N + \sum_{i \in I} c_i \lambda_i X_i \tag{1}$$

$$\text{subject to } N - \sum_{i \in I} r_i \lambda_i X_i \geq k \tag{2}$$

$$\lambda_i = m_i + \sum_{j \in \Gamma_i} \lambda_j (1 - X_j), \forall i \in I \tag{3}$$

$$\lambda_i \leq \lambda_i X_i + \sum_{j \in A_i} \lambda_j X_j, \forall i \in I \tag{4}$$

$$N \in \mathbb{N} \tag{5}$$

$$X_i \in \{0, 1\}, \forall i \in I \tag{6}$$

**Table 1**

Asset structure settings: numbers of components per indenture level.

PS	Parameter			Indenture level					
	$ \Gamma_0 $	$ \overline{\Gamma}_i $	$ L $	1	2	3	4	5	6
1,3	50	2	3	50	100	200	-	-	-
	100	2	3	100	200	400	-	-	-
	50	4	3	50	200	800	-	-	-
	100	4	3	100	400	1600	-	-	-
2	50	2	3	50	100	200	-	-	-
	50	2	4	50	100	200	400	-	-
	50	2	5	50	100	200	400	800	-
	50	2	6	50	100	200	400	800	1600

Constraint 2 is a constraint on the number of assets that must be available for operation. Constraint 3 is the recursion stating that an item  $i \in I$  is replaced both when failing individually ( $m_i$ ), and upon failure of one of its child items that have not been defined as LRU. Finally, Constraint 4 assures that a child non-LRU that fails must have an LRU ascendant. If  $\lambda_i = 0$  or if  $X_i = 1$ , the constraint is always satisfied. If  $X_i = 0$  and  $\lambda_i > 0$ , then some parent item must be defined as LRU, i.e.,  $\sum_{j \in A_i} X_j \geq 1$ , and thus  $\lambda_i \leq \sum_{j \in A_i} \lambda_j X_j$ .

### 5. Numerical Experiment

This section explains our numerical experiment. We first give the setup of our numerical experiment in Section 5.1; a detailed description of how we generate some parts of the problem instances can be found in Appendix C. We show the results in Section 5.2. The LRU definition problem is implemented using the CPLEX 12.6 Class API for MATLAB R2013a on an Intel Core i5 M540@2,53 gigahertz, with 4 gigabytes RAM running 64-bit Windows 7. All instances are solved to optimality; we do not use a time limit.

#### 5.1. Instance generator

We use a problem instances generator that is inspired by that of Basten et al. (2012). We define three problem sets (PS): PS1, PS2 and PS3. Table 1 lists the settings used to modify the asset structures for each of the three sets, and Table 2 lists the settings for the other parameters. Typical values found at NedTrain are in between the low and high values used in the problem sets. We use a full factorial design per PS, and generate ten problem instances per parameter setting to avoid basing conclusions on one unique instance only.

Since maintenance is a labor intensive service, in our experiments we explicitly consider the effect of wages on the LRU definitions. We define the labor cost as the product of the yearly wage,  $w$ , and the direct labor time,  $r_i$ . This means that the cost of replacement of an LRU  $i \in I$  is:  $c_i = c'_i + wr_i$ , with  $c'_i$  covering material and indirect labor cost.

We use three parameters to produce the asset structures used in each of the three problem sets: (i) the number of items in the first indenture level, denoted by  $|\Gamma_0|$ , the (ii) average number of items per parent, denoted by  $|\overline{\Gamma}_i|$  and (iii) the number of indenture levels, denoted by  $|L|$ .

In PS1 and PS2 we explore the effect of different indenture structure combinations (see Table 1). In PS1 we fixate the number of indenture levels while varying the average number of items per parent,  $|\overline{\Gamma}_i|$ , and the number of items in the first indenture,  $|\Gamma_0|$ . In PS2 we vary the number of indenture levels,  $|L|$ , while keeping constant  $|\Gamma_0|$  and  $|\overline{\Gamma}_i|$ . Besides the different indenture structure combinations, each problem instance in PS1 and PS2 takes one value or range for each of the seven parameters from Table 2. This means that we generate a total of  $10 \times 2^7 \times 4 = 5,120$  problem instances per PS.

Based on the results of PS1 and PS2, PS3 has been designed to focus on the effect of changes in the parent cost,  $c'_i \mid \Gamma_i \neq \emptyset$ , as a func-

**Table 2**  
Parameters settings or sampling range for PS1, PS2 and PS3.

Parameter		Setting	
		1	2
Item failure rate per asset (1/year)	$m_i/k$	[0.01; 0.1]	[0.01; 1]
Item replacement lead time (hour) <sup>a</sup>	$r_i$	[0.25; 2]	[0.5; 4]
Item replacement cost (euro × 1000)	$c_i^b$	[1; 10]	[10; 100]
Parent-child cost factor	$f_c$	[0.5; 1.5]	[1; 3]
# Required operational assets <sup>a</sup>	$k$	10	100
Asset cost (euro × 1000)	$c_0$	[200; 400]	[1000; 2000]
Wage (euro/hour)	$w$	5.5	55

<sup>a</sup> For PS3  $r_i$  and  $k$  are fixed at setting 1.

<sup>b</sup> For leaf items  $i \in I \mid \Gamma_i = \emptyset$ .

tion of the costs of its children. Therefore, the setup is identical to that of PS1, with the exception of two parameters that remain fixed at setting 1 (see Table 2) because they were found not to influence the results in PS1 or PS2. This means that for PS3 we generate  $10 \times 2^5 \times 3 \times 4 = 3,840$  problem instances.

We give more details on the parent-child cost factor,  $f_c$ , and the parent cost,  $c_i \mid \Gamma_i \neq \emptyset$ , because this helps to explain the results in Section 5.2. A detailed explanation of the other parameters is given in Appendix C. For PS1 and PS2 the cost of a parent assembly is proportional to the cost of its most expensive child item, i.e.,  $c_i = f_c \max_{j \in \Gamma_i} c_j \mid j \in \Gamma_i$ , for  $i \in I \mid \Gamma_i \neq \emptyset$ . In addition to this setting, in PS3, we also consider  $c_i = \frac{f_c}{|\Gamma_i|} \sum_{j \in \Gamma_i} c_j$  and  $c_i = f_c \sum_{j \in \Gamma_i} c_j$ . The former setting means that a parent assembly has the average cost of the child items,  $j \in \Gamma_i$ , multiplied by a cost factor. In the latter setting, the cost of a parent is the sum of the cost of the child items multiplied by the cost factor.

5.2. Results

Computation times are mainly driven by the changes in the asset structure settings of Table 1. Therefore, Table 3 shows the minimum, mean and maximum computation times for each subset of problem instances that share the same asset structure settings. The computation times are typically low, so we do not further focus on them. The remainder of this section discusses the results by answering the following questions:

1. What cost increases result when we compare the optimal LRU definition with defining (i) the first indenture (largest) items as LRU and (ii) the highest indenture (smallest) items as LRUs?
2. Which model parameters influence the cost increases in the above cases?

To answer these questions, we solve each problem instance to find the optimal LRU definition, and we apply two heuristics that are based on what we have seen in practice. We call the first heuristic

the *large heuristic* (make LRUs as large as possible, i.e., all first indenture items), and denote the resulting costs by  $C^l$ . We call the second heuristic the *small heuristic* (make LRUs as small as possible, i.e., all leaf items), and the resulting costs are denoted by  $C^s$ . The optimal costs are denoted by  $C^*$ . Notice that both heuristics represent a naive approach, because they are generally not followed for all items when defining LRUs in practice. However, we believe that it is a good reference point for contrasting extreme cost results.

We answer Question 1 by showing the cost increases that appear when comparing  $C^*$ , with the cost of the two heuristic solutions:  $C^l$  and  $C^s$ . The percentage increase that we show is calculated as  $\frac{C^l/s - C^*}{C^*}$ . Question 2 addresses the influence of model parameters on the possible cost increases. Answering this question gives managerial insights about the relevant cost drivers in the model. We use an  $n$ -way analysis of variance to test the significance of the results for different parameter settings.

To answer Question 1, Table 4 summarizes the cost increases over all problem instances. We see that for all PSs, using the heuristics results in huge cost increases compared to the optimal solution, with the small heuristic performing better than the large heuristic. The cost increase for all problem instances is at least 4 percent (shown in Appendix E). These results show that solving the LRU definition problem considering technical aspects only, as is currently common in practice (see Section 3), typically leads to high additional costs. Optimization incorporating economic criteria is required.

We next answer Question 2. We discuss only those parameters that have a significant effect on the achieved cost increase. Item replacement time,  $r_i$ , and the target number of assets,  $k$ , had no significant effect on the cost increase in any of the PSs. Fig. 5 shows the main effects plot of the cost increase in PS1 for those parameters that have a significant effect ( $p$ -values less than 0.05 in the  $n$ -way analysis of variance of the cost increase). We discuss each of these effects below. Appendix E shows the minimum (*min*), average (*mean*) and maximum (*max*) cost increase for parameter settings of PS1, PS2 and PS3.

The largest cost increases result from increasing the number of items per parent ( $|\Gamma_i|$  from 2 to 4) and the cost factor ( $f_c$  from the range [0.5; 1.5] to the range [1; 3]). Both increases mean that costs of leaf items remain the same while parent items become more expensive. As a result, it becomes more costly to make a suboptimal

**Table 3**  
Computation times of our experiments (in seconds).

PS	Parameter			Computation time		
	$ \Gamma_0 $	$ \Gamma_i $	$ L $	min	mean	max
1	50	2	3	0.01	0.03	0.20
	100	2	3	0.03	0.05	0.21
	50	4	3	0.04	0.07	0.26
	100	4	3	0.12	0.18	0.46
2	50	2	3	0.01	0.03	0.24
	50	2	4	0.03	0.06	0.33
	50	2	5	0.07	0.15	0.48
	50	2	6	0.20	0.33	18.37
3	50	2	3	0.03	0.04	0.22
	100	2	3	0.03	0.05	0.08
	50	4	3	0.05	0.06	0.07
	100	4	3	0.11	0.16	0.39

**Table 4**  
Average cost increase between optimal and heuristic solution for PS1–PS3.

Problem set	# Instances	Cost increase	
		$C^l$ vs $C^*$ (percent)	$C^s$ vs $C^*$ (percent)
PS1	5120	62	33
PS2	5120	80	54
PS3	3840	112	33

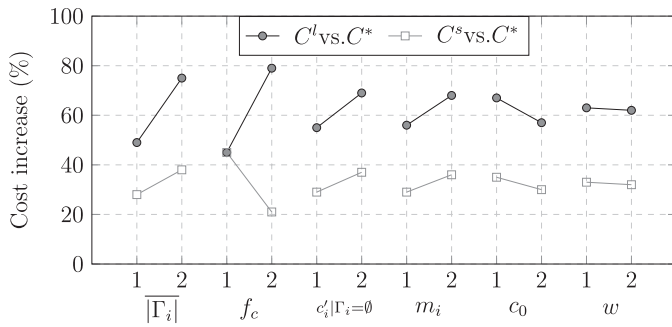


Fig. 5. Percentage cost increase for settings (1 or 2) of model parameters for PS1.

LRU definition. Furthermore, in the optimal solution, the number of LRUs doubles and LRUs will be smaller than before. Therefore, we see that the percentage cost increase of the large heuristic compared with the optimal solution becomes much higher, while this effect is much smaller ( $|\Gamma_i|$ ) or even reversed ( $f_c$ ) for the small heuristic.

Increasing the item replacement cost,  $c_i | \Gamma_i = \emptyset$ , or the item failure rate,  $m_i$ , leads to a slightly higher cost increase. While these setting changes do not significantly influence the number of LRUs in the optimal solution, the LRU replacement costs increase with increases in  $c_i$  or  $\lambda_i$ . Therefore, the relative contribution of the costs of the fleet to the total optimal costs becomes much smaller: it becomes more costly to make a suboptimal LRU definition.

With increasing asset price,  $c_0$ , we found an average reduction in the cost increase for both heuristics. Increasing  $c_0$  has the inverse effect of  $c_i$  or  $\lambda_i$ . The reduction in the cost increase results because for more expensive assets the costs of the fleet are higher, and the relative weight of the LRU replacement costs becomes smaller. Finally, changes in  $w$  do not lead to large cost increases. The reason is that wages represent a relatively small contribution to the total costs.

Fig. 6 shows the main effects plot of the cost increase due to an increase in the number of indenture levels,  $|L|$ , from three to six indentures (settings 1–4) for PS2. Increasing  $|L|$  creates more opportunity for allocating cheaper repair options in the optimal solution. We find that the number of LRUs in the optimal solution has large variations within each setting, while remaining almost unchanged for both heuristics, e.g., for  $|L| = 6$ , problem instances range from 478 LRUs (closer to the large heuristic) to 1476 LRUs (closer to the small heuristic). Both heuristics result in increased costs.

Fig. 7 shows the main effects plot of the cost increase given the parent item cost settings for PS3. The differences between the average cost of the parent between settings  $\frac{\sum_{j \in \Gamma_i} c_j}{|\Gamma_i|}$  and  $\max c_j$  is very small, which explains why they result in very similar cost increases. With  $\sum_{j \in \Gamma_i} c_j$ , more small items are defined as LRU in the optimal solution. This makes a small cost increase for the small heuristic, but makes a huge cost increase for the large heuristic.

Summarizing, the cost of parent assemblies relative to child items can have a huge influence on the cost increases. Therefore, if parents

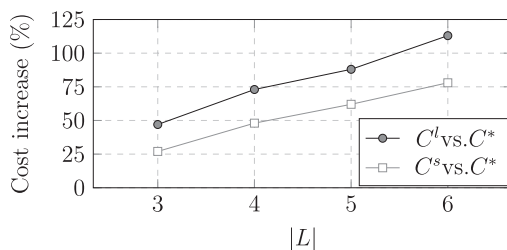


Fig. 6. Average percentage cost increase for increasing indenture levels of PS2.

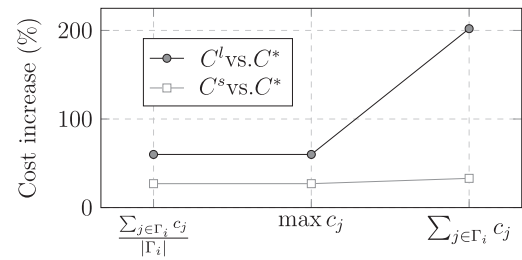


Fig. 7. Median percentage cost increase for settings of parent cost,  $c_i | \Gamma_i \neq \emptyset$ , for PS3.

become more expensive compared to their children, there is more to gain by making optimal LRU definitions, and these optimal definitions include smaller items.

### 6. Conclusions

In this paper we have discussed a problem within maintenance optimization: the optimal selection of replacement level within the indenture structure of capital assets. To the best of our knowledge, this problem has not received attention in the literature. We have presented the LRU definition problem as a decision on whether or not to replace a parent assembly or a child item, upon failure of that child item. The LRU definition problem explicitly accounts for the structural dependency existing between items within the indenture structure of capital assets. In practice, the LRU definition decision is usually made ad-hoc, or based only on engineering/technical criteria. Furthermore, many maintenance repair decisions, such as the Level of Repair Analysis decisions, assume the LRU definitions implicitly.

We have shown that the LRU definition problem is NP-hard, that it can be optimized using an MILP model, and that significant cost reductions can be achieved when compared to two heuristics commonly used in practice. This provokes further research about the ways to improve LRU decisions in practice. Some of the assumptions in the current model of the LRU definition problem may be relaxed, especially that of sequential maintenance.

The model can also be extended by incorporating limited labor capacity. We have seen that wage differences do not significantly influence in LRU definitions. However, we have also seen in practice a growing concern for labor scarcity in the maintenance industry. Limited manpower (working hours per skill sets) could be an additional constraint in the model. Increasing task frequency builds routine, and helps prevent costly mistakes in maintenance work. A lower bound on replacement frequency can be added as constraint.

Proceeding in the direction of combining the LRU definition problem with other problems would be useful to support the joint maintenance and logistics decisions for re-defining LRUs, for example at NedTrain. One way would be to combine it with LORA, e.g., the work by Basten et al. (2011). This means combining the MILP model for the LRU definition problem with the MILP model for the LORA problem. It may also be beneficial to help maintenance assessment by combining the LRU definition problem with the joint optimization of level of repair analysis (LORA) and spare parts stocking (see Section 2.4). One way would be by extending the work by Basten et al. (2012). This means combining the MILP model for the LRU definition problem with the MILP model for the LORA problem. Solving that MILP first and then a spare parts stocking model. Using a feedback loop to the MILP model leads to an iterative heuristic to solve the joint problem.

### Acknowledgments

This research is part of the “Rolling Stock Life Cycle Logistics” applied research and development program, funded by NS/NedTrain.



We thank the experts at Thales and NedTrain for their invaluable help and insights. The second author gratefully acknowledges the support of the Lloyd’s Register Foundation (LRF). LRF helps to protect life and property by supporting engineering-related education, public engagement and the application of research. We gratefully acknowledge the editor and the reviewer for their valuable comments that helped improve this paper.

**Appendix A. Linearization**

In this appendix, we linearize the mathematical model described in Section 4.2. The objective function, and Constraints 2–4, contain products of the cumulative failure rate  $\lambda_i$  and the decision variable  $X_i$ . This product is a nonlinear expression. We linearize these expressions by introducing an auxiliary variable  $\rho_i$  that is equal to  $\lambda_i X_i$ . We further define  $\Lambda_i$  as the failure intensity of item  $i \in I$ , including the failure intensity of all its descendants. It is a Big M variable and it is computed recursively:  $\Lambda_i = m_i + \sum_{j \in \Gamma_i} \Lambda_j$ . The linearized LRU definition problem is shown below. The objective function and constraints A.2 to A.4 are the same as the objective function and constraints 2 to 4 in the original model, with  $\lambda_i X_i$  being replaced by  $\rho_i$ . Constraints A.5 to A.8 are added.

$$\text{minimize } c_0 N + \sum_{i \in I} c_i \rho_i \tag{A.1}$$

$$\text{subject to } N - \sum_{i \in I} r_i \rho_i \geq k \tag{A.2}$$

$$\lambda_i = m_i + \sum_{j \in \Gamma_i} (\lambda_j - \rho_j), \forall i \in I \tag{A.3}$$

$$\lambda_i \leq \rho_i + \sum_{j \in A_i} \rho_j, \forall i \in I \tag{A.4}$$

$$\rho_i \leq \Lambda_i X_i, \forall i \in I \tag{A.5}$$

$$\rho_i \leq \lambda_i, \forall i \in I \tag{A.6}$$

$$\rho_i \geq \lambda_i - \Lambda_i (1 - X_i), \forall i \in I \tag{A.7}$$

$$\rho_i \geq 0, \forall i \in I \tag{A.8}$$

$$N \in \mathbb{N} \tag{A.9}$$

$$X_i \in \{0, 1\}, \forall i \in I. \tag{A.10}$$

**Appendix B. Proof that the LRU definition problem is NP-hard**

We show that the binary knapsack problem (BKP) can be reduced to the LRU definition problem in polynomial time. Since the BKP is known to be NP-hard, see for example Martello and Toth (1990, p. 6), this proves that also the LRU definition problem is NP-hard.

The BKP is the problem of selecting items  $j \in J$  to put in a knapsack with a fixed capacity  $C$ , such that a profit is maximized. Each item  $j \in J$  gives a profit  $p_j$  and has a weight  $w_j$ . An item  $j \in J$  is selected to be put in the knapsack if  $x_j = 1$ ;  $x_j = 0$  otherwise. The mathematical model

formulation is then

$$\begin{aligned} &\text{minimize } \sum_{j \in J} p_j x_j \\ &\text{subject to } \sum_{j \in J} w_j x_j \leq C \\ &x_j \in \{0, 1\}, \forall j \in J \end{aligned}$$

We next construct an instance of the LRU definition problem. We set  $k = 1$  and below we will show that as a result of our formulation,  $N = 2$ . For now, we thus assume that  $k - N = -1$ . We define a product structure with one dummy component, indicated by  $-1$ , that has  $|J|$  subcomponents; to be more precise,  $\Gamma_{-1} = J$ . We define  $p^{\max} = \max_{j \in J} \{p_j\}$  and set  $c_{-1} = p^{\max}$  and  $m_{-1} = 0$ . We then set, for  $i \in J$ ,  $m_i = 1$ ,  $c_i = p^{\max} - p_i$ , and  $r_i = w_i/C$ . We come back to  $r_{-1}$ , but we assume for now that  $r_{-1} = 0$ . The LRU definition problem for this instance is then:

$$\text{minimize } 2c_0 + p^{\max} \lambda_{-1} + \sum_{i \in J} (p^{\max} - p_i) X_i \tag{B.1}$$

$$\text{subject to } \sum_{i \in J} \frac{w_i}{C} X_i \leq 1 \tag{B.2}$$

$$\lambda_{-1} = \sum_{i \in J} (1 - X_i)$$

$$\lambda_i = 1, \forall i \in J \tag{B.3}$$

$$\lambda_i \leq X_i + \lambda_{-1} X_{-1} \tag{B.4}$$

$$X_i \in \{0, 1\}, \forall i \in J \cup \{-1\} \tag{B.5}$$

To see that the objective function is correct, there are two things to notice. Firstly, notice that we omit the term  $X_{-1}$  from the objective function because  $X_{-1} = 0 \iff \lambda_{-1} = 0$ , while  $X_{-1} = 1$ , otherwise. Secondly, notice that we omit the term  $\lambda_i$  from the objective function because  $\lambda_i = 1$  for  $i \in J$  as a result of Constraint (B.3). Using that  $\lambda_{-1} = \sum_{i \in J} (1 - X_i)$ , following Constraint (B.3), and that  $X_i \in \{0, 1\}$ , we can rewrite the objective function to become:

$$\text{minimize } 2c_0 + p^{\max} |J| - \sum_{i \in J} p_i X_i,$$

and since the first two terms are constant, this is equivalent to:

$$\text{maximize } \sum_{i \in J} p_i X_i.$$

By further noticing that in Constraint (B.2) we can multiply both sides of the equation by  $C$  and that Constraints (B.3) and (B.4) are basically used for book keeping, we see that we have constructed an instance of the LRU definition problem that, when solved, gives the solution for our original binary knapsack problem. (Notice that to find Constraint (B.2) we have used that we can make both sides positive instead of negative, which means that the inequality is reversed.)

The only point remaining is that we need to ensure that  $N = 2$ . We can achieve this by, on the one hand, ensuring that there is always a strictly positive (small) workload, so that  $N > k = 1$ , and on the other hand, setting  $c_0 \geq p^{\max} |J|$  so that  $N \leq 2$ . Ensuring that there is always a small workload can be achieved by setting  $r_{-1} = \epsilon$  and setting  $r_i = \epsilon + w_i(1 - \epsilon)/C$ , with  $\epsilon |J| \ll \min_{i \in J} \{w_i\}$ , and  $0 < \epsilon |J| < 1$ . Instead of Eq. (B.2) above, Eq. 2 in the LRU definition model will then become

**Table D.5**  
LRU requirements at Thales.

Criteria	An LRU shall be...
<b>Mandatory Criteria</b>	
Structure	Physically separable from the asset
Identification	Uniquely identifiable for codification as a spare part
Testing	Independently testable and reproducible
Task	Replaced using prescribed maintenance procedures
<b>Preferred Criteria</b>	
Fault detection	Detected and localized by use of built-in testing
Handling	Sized not to exceed the following limits (95th percentile, weighed with the failure rates <sup>a</sup> ): weight of one LRU is lower than 16 kilograms, dimensions (millimeter) of one LRU are lower than 600 × 450 × 450. Remaining LRUs (max. 5th percentile) shall be in accordance with the requirements of MIL-STD-1472F, §5.9.11
Weight	Of a target weight of less than 10 kilograms. If the LRU weight is between 10 kilograms and 25 kilograms, provisions shall be made for two-handed transportation. If the weight is between 25 kilograms and 45 kilograms, transportation by two persons or hoisting facilities shall be in place. If the LRU weight exceeds 45kg, the LRU shall be equipped with hoisting facilities
Interface	Secured in place by mechanical fastening (clip or screw) to a maximum of 4
Interface (electric)	Connected via free mounted plug and socket connectors, to a maximum of 3
Interface (fluid)	Provided with “quick disconnect” type liquid or air connections
Reliability	Every equipment, assembly, or part contributing to the asset MTBF with a failure rate of more than 2 per million hours
Task	Replaced within a time of 30 minutes (hands-on-tool-time 95th percentile, weighed with the failure rates <sup>a</sup> )
Adjustment	Pre-tuned, so that it can be replaced without readjustment

<sup>a</sup> At least 95 percent of all maintenance actions should conform to this requirement.

(notice that  $\lambda_{-1} = \sum_{i \in J} (1 - X_i)$ ):

$$N - \sum_{i \in J} \left[ \epsilon + \frac{w_i(1 - \epsilon|J|)}{C} \right] X_i - \epsilon \sum_{i \in J} (1 - X_i) \geq k$$

$$\epsilon|J| + \sum_{i \in J} \frac{w_i(1 - \epsilon|J|)}{C} X_i \leq N - k,$$

which ensures that there is always a small positive workload, while the same optimal solution will still result.

**Appendix C. Problem instance generator**

In this appendix, we explain the problem instances generator. The settings that we use for the parameters are listed in Table 2 in Section 5.1. In the explanation here, we assume setting 1 for each parameter. The number of items per indenture level are given in Table 1 in Section 5.1. Each item at indenture level  $L + 1$  is randomly assigned to a parent (at indenture level  $L$ ) using a uniform distribution on the parent indenture’s index range. The number of children per parent will thus differ per parent. The annual demand for an item,  $m_i$ , is drawn per item from a uniform distribution on the interval  $[0.01; 0.1]$ . The leaf item’s replacement costs, i.e.,  $c_i$ , for  $i \in I \mid \Gamma_i = \emptyset$ , are drawn from a shifted exponential distribution with shift factor 1000 and rate parameter  $7/(10,000 - 1000)$ . As a result, we do not have items with a price below 1000 euros and approximately 5 percent of the items’ prices exceed 10,000 euros. Therefore, there are typically more cheap items than expensive items. The parameter  $f_c$  is drawn from a uniform distribution on the range  $[0.5; 1.5]$ . The item replacement lead time per hour,  $r_i$ , is drawn from a uniform distribution on the range  $[0.25; 2]$ . Asset prices are drawn from a uniform distribution on the range  $1000 \text{ euros} \times [200; 400]$ .

**Appendix D. LRU requirements at Thales: mandatory and preferred criteria**

Table D.5 shows a summary of the LRU requirements at Thales, including mandatory and preferred criteria.

**Appendix E. Results summary**

Table E.6 shows the results for PS1, PS2 and PS3. We only show the full list of results for parameter settings in PS1, because these parameter settings had similar results in PS2 and PS3.

**Table E.6**  
Cost increases for settings of parameters in all three problem sets.

PS	Par.	Set.	C <sup>1</sup> vs C* (percent)			C <sup>2</sup> vs C* (percent)		
			min	mean	max	min	mean	max
1	Γ <sub>0</sub>	1	5	62	132	4	32	68
		2	10	62	117	7	33	64
	$\overline{ \Gamma_i }$	1	5	49	113	4	28	61
		2	19	75	132	11	38	68
	$f_c$	1	5	45	81	7	45	68
		2	14	79	132	4	21	33
	$c'_i$	1	5	55	130	4	29	66
		2	21	69	132	12	37	68
	$m_i$	1	5	56	131	4	29	66
		2	19	68	132	11	36	68
	$r_i$	1	5	62	132	4	33	68
		2	5	62	132	4	33	68
$k$	1	5	62	132	4	33	68	
	2	5	62	132	4	33	68	
$c_0$	1	15	67	132	10	35	68	
	2	5	57	131	4	30	67	
$w$	1	5	63	132	4	33	68	
	2	5	62	132	4	32	68	
2	J	1	5	47	113	4	27	61
		2	13	73	133	12	48	86
		3	28	88	141	25	62	97
3	$c'_i$ <sup>a</sup>	4	39	113	179	34	78	126
		1	5	60	132	4	27	68
		2	5	60	132	4	27	68
		3	24	202	423	5	33	61

<sup>a</sup> Setting 1, 2 and 3 are  $c'_i = \frac{f_c}{|\Gamma_i|} \sum_{j \in \Gamma_i} c'_j$ ,  $c'_i = f_c \max_j c'_j \mid j \in \Gamma_i$ , and  $c'_i = f_c \sum_{j \in \Gamma_i} c'_j$ , respectively, for all  $i \in I \mid \Gamma_i \neq \emptyset$ .

**References**

Barabadi, A., Barabady, J., & Markeset, T. (2011). Maintainability analysis considering time-dependent and time-independent covariates. *Reliability Engineering & System Safety*, 96(1), 210–217.

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

Basten, R., & van Houtum, G. (2014). System-oriented inventory models for spare parts. *Surveys in Operations Research and Management Science*, 19(1), 34–55.

Basten, R. J. I., van der Heijden, M. C., & Schutten, J. M. J. (2011). A minimum cost flow model for level of repair analysis. *International Journal of Production Economics*, 133(1), 233–242.

Basten, R. J. I., van der Heijden, M. C., & Schutten, J. M. J. (2012). Joint optimization of level of repair analysis and spare parts stocks. *European Journal of Operational Research*, 222(3), 474–483.

- Blanchard, B. S., & Fabrycky, W. J. (2011). *Systems engineering and analysis* (5th ed.). Upper Saddle River: Prentice Hall.
- Dhillon, B. S., & Liu, Y. (2006). Human error in maintenance: a review. *Journal of Quality in Maintenance Engineering*, 12(1), 21–36.
- DoD (1996). *MIL-PRF-49506 performance specification logistics management information*. Washington: DoD (Department of Defense).
- Jensen, P. A. (1975). Optimum repair level analysis: some extensions. In M. A. Geisler (Ed.), *Logistics* (pp. 87–104). Amsterdam: North-Holland/American Elsevier.
- Jones, J. V. (2006). *Supportability engineering handbook: implementation, measurement, and management*. New York: SOLE Logistics Press/McGraw-Hill.
- Lambert, A. J. D. (2003). Disassembly sequencing: A survey. *International Journal of Production Research*, 41(16), 3721–3759.
- Marais, K. B., Rivas, J., Tetzloff, I. J., & Crossley, W. a. (2013). Modeling the impact of maintenance on naval fleet total ownership cost. In *IEEE international systems conference (syscon)* (pp. 801–808). Orlando, FL: IEEE.
- Martello, S., & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. 1990. Chichester, UK: John Wiley & Sons.
- Moreu De Leon, P., González-Prida Díaz, V., Barberá Martínez, L., & Crespo Márquez, A. (2012). A practical method for the maintainability assessment in industrial devices using indicators and specific attributes. *Reliability Engineering & System Safety*, 100, 84–92.
- Moubray, J. (1997). *Reliability-centered maintenance*. New York: Industrial Press.
- Muckstadt, J. A. (2005). Analysis and algorithms for service parts supply chains. *Springer Series in Operations Research and Financial Engineering* (2nd ed.). Berlin: Springer.
- Nicolai, R. P., & Dekker, R. (2008). Optimal maintenance of multi-component systems: a review. In D. Murthy, & K. Kobbacy (Eds.), *Complex system maintenance handbook* chapter 11 (pp. 263–286). London: Springer London.
- Sasieni, M. W. (1956). A Markov chain process in industrial replacement. *Operations Research*, 7(4), 148–155.
- Schotborgh, W. O., McMahon, C., & Houten, F. J. V. (2012). A knowledge acquisition method to model parametric engineering design processes. *International Journal of Computer Aided Engineering and Technology*, 4(4), 373–391.
- Sherbrooke, C. C. (2004). Optimal inventory modeling of systems: multi-echelon techniques. *International series in operations research & management science* (2nd ed.). London: Kluwer Academic.
- Thomas, L. C. (1986). A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering*, 16(4), 297–309.
- Tinga, T. (2010). Application of physical failure models to enable usage and load based maintenance. *Reliability Engineering & System Safety*, 95(10), 1061–1075.