Life cycle costs optimization for capital goods

Life Cycle Costs Optimization for Capital Goods

This thesis is part of the Ph.D. thesis series of the Beta Research School for Operations Management and Logistics (onderzoeksschool–beta.nl) in which the following Dutch universities cooperate: Eindhoven University of Technology, Maastricht University, University of Twente, VU Amsterdam, and Wageningen University and Research.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-4479-0

This research has been funded by The Netherlands Organisation for Scientific Research, ASML, and Dutch Railways.

Life Cycle Costs Optimization for Capital Goods

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. F.P.T. Baaijens, voor een
commissie aangewezen door het College voor
Promoties, in het openbaar te verdedigen
op donderdag 21 juni 2018 om 16:00 uur

door

Joni Peter Carolus Driessen

geboren te Helden

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. I.E.J. Heynderickx
1e promotor:	prof.dr.ir. G.J.J.A.N. van Houtum
co–promotor:	dr.ir. J.J. Arts
leden:	prof.dr.ir. R. Dekker (Erasmus Universiteit Rotterdam)
	prof.dr. T. van Woensel
	prof.dr. G.J. Woeginger (RWTH Aachen University)
adviseur:	prof.dr. A.A. Scheller–Wolf (Carnegie Mellon University)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgments

Going back to the start of my university trajectory in 2007/2008, I – and numerous others that already knew me by then – would have not imagined me starting and finishing a PhD. How different everything turned out in those passed 11 years, as this thesis proves. Moments of struggle, frustration, joy, cooperation and pride have succeeded one another in the past four years. Most of these moments are the result of the interaction with many people, who each have contributed to the realization of this thesis. For that, I deeply thank each and every one of you. However, there are some people that deserve extra attention.

First of all, I am very grateful to my co–promotor and daily supervisor Joachim Arts. Your enthusiasm, positive attitude towards research problems, and extensive knowledge of mathematical techniques has been of tremendous guidance in the last four years. I thank you for the critical remarks and comments, for the spontaneous and lengthy meetings, for discussions on less relevant things (like what fonts look nice in a thesis), and for your time and effort invested in me.

Next, I would like to express my gratitude to Geert–Jan van Houtum. Your ability to dissect and frame problems is remarkable, from time to time frustrating, but in the end one of the most valuable things I have learned from you. I also thank you for stimulating me to zoom out and to rethink what research means in a practical context.

It took me more than a year and a half to prove Theorem 3.1 in Chapter 3. This proof would not have made it into this thesis if it were not for Joost de Kruijff. I remember telling you about the problem that I was facing with "this annoying thing that *has* to be true" back in 2017. A couple of days later we had discussed the problem

extensively to conclude that “it is pretty annoying to prove” (your own words, Joost). Yet, despite its annoyance, we managed to crack the problem and I am very grateful for all your help and input.

Then, a special word for Alan Scheller–Wolf. I would like to thank you for hosting my visit to Carnegie Mellon University in the spring of 2017. Despite a busy agenda, you always gave me the impression it was empty during our discussions on research. I thank you for our cooperation that resulted in Chapter 5, and I am very glad that you agreed to be part of my committee.

I would also like to thank Rommert Dekker, Gerhard Woeginger, and Tom van Woensel for being on my thesis committee, taking the time to read and review the thesis, and providing me with valuable feedback.

It has been a great pleasure to conduct part of my research at the two project partners, ASML and Dutch Railways. I would like to thank both parties for their support. In particular, I would like to thank Mehmet Atan for the feedback from a practical perspective and René Habets for the inspiration and fruitful discussions related to Chapters 3 and 4. At Dutch Railways, I am very grateful to Bob Huisman for his open minded attitude towards my research problems and the excellent ability to relate these problems to Dutch Railways.

Doing a PhD is rather lonesome job, but luckily I have shared an office with two nice roommates, who both deserve some extra attention. I would like to thank my first roommate Denise Tönissen for providing a nice working atmosphere in our office and for the nice discussions on research and non–research topics. I also thank my second roommate Simon Voorberg for his company and sense of humor during the last year of my PhD.

It has been a pleasure work in the OPAC group, and this is a direct consequence of the nice atmosphere created by all colleagues. Specifically, I would like to thank Zümbül Atan, Sjors Jansen, Kay Peeters, Yeşim Koca, Mirjam Meijer, and Simon Voorberg for joining me to the canteen for a morning/afternoon coffee. Furthermore, I would like to thank Claudine Hulsman and Christel van Berlo for the very welcome social moments, each of which I deeply value.

Next, I thank my friends for the interest in my research, even if it were only pretended interest. However, I am particularly thankful for the provided distraction from research. The lunches and dinners together, as well as the weekend/holiday trips have truly been valuable for taking my mind off research.

Finally, a brief word for my closest family. I am very grateful to the unconditional interest of my parents in my research. However, this stands in pale comparison to your support and the life lessons you have taught me. These have strongly influenced the successful completion of this thesis; thank you for all of this. A last and most important word belongs to Guusje. Your patience and understanding for a drifting mind are most admirable, and have been instrumental to this thesis. Your support

and love, however, have been significantly more important for completing this thesis. Thank you.

Contents

1

Introduction

The purchase of a new car is not an overnight decision. Many people compare various cars over and over again until they reach a decision on which car to invest. The investment decision is complex: it is not based on a single dimension of the car such as the price. Instead, many people compare various cars based on multiple dimensions: purchase price, depreciation, fuel economy, insurance costs, reliability estimates, road tax[1] etc. In other words, the total cost of owning each of the various cars are compared, and subsequently a well considered investment decision is made. The same decision process is followed in many other settings, but it is particularly useful when the costs of owning a system become large. For such capital intensive systems – e.g. aircraft, trains, lithography systems, MRI scanners, and baggage handling systems – the operational costs can account for 70–80% of the total costs of ownership[2] (Saranga and Kumar, 2006; Öner et al., 2007; van Dongen, 2011). Therefore, considering other aspects besides the initial purchase price is crucial for making the right investment decision. As a consequence, customers of capital intensive systems often require Original Equipment Manufacturers (OEMs) to offer a complete package that considers the total costs over the product's lifetime including the initial purchase cost, the operational costs and the costs of retiring a system – rather than just the initial purchase costs. Some examples of such customer – OEM relationships are: KLM Royal Dutch Airlines (customer) – Airbus (OEM), Samsung (customer) – ASML (OEM), Dutch Railways (customer) – Bombardier (OEM). In response, OEMs have developed contractual mechanisms that make them responsible for more than just design and production: OEMs have started to take responsibility for a so–called

[1] Car owners pay an annual road tax in The Netherlands.

[2] Authors typically include the costs for downtime in these figures.

system life cycle.

This thesis deals with decisions that affect this life cycle and the corresponding costs. As a consequence, we first explore a system's life cycle in further detail in Section 1.1. Moreover, we shed some more light on the responsibilities that the OEM carries throughout a system's life cycle. Subsequently, we explain – in Section 1.2 – three different decision problems that the OEM encounters during the life cycle of a system. Each of these problems influence the costs incurred later in the life cycle and are not trivially solved. Therefore, our objective is to develop mathematical decision support models that help to solve the decision problems. This research objective is presented in Section 1.3. In pursuing this research objective, we make various contributions that are presented in Section 1.4. We conclude this chapter in Section 1.5 presenting an outline and the main methodologies used in each of the following chapters.

1.1. A system life cycle

The identification of the needs and requirements initiates the life cycle of a system. The system is designed carefully, after which it is produced. Subsequently, the system is exploited during the usage phase; the OEM or user may modify the system during this usage phase to enhance performance. Finally, the system ages and reaches its retirement at some point in time. We have depicted a system life cycle in Figure 1.1, based on Kumar et al. (2012).

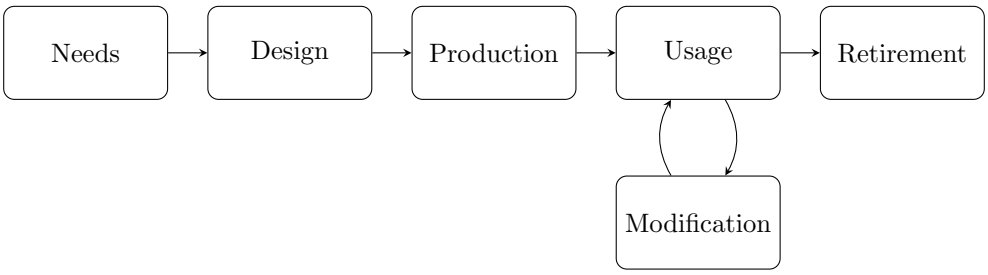


Figure 1.1: A system life cycle

Traditionally, OEMs were only concerned with the identification of the needs, the design, and production of their systems. After production, the systems were sold to the customer and all operational aspects would be the customer's responsibility. However, recently the contracts that OEMs close with their customers incentivize the OEMs to consider aspects from the usage phase (also called after–sales aspects) such as service part provisioning, maintenance and system modifications. Two common contracts are full service contracts and performance based contracts (Cohen et al., 2006; Selviaridis and Wynstra, 2015). Under a full service contract, the customer pays the OEM a periodic fee and the OEM has to ensure that his systems meet

certain criteria such as availability or productivity criteria. This means that the OEM performs maintenance to the systems, but he also carries service part inventories in order to respond quickly to system failures. In a more extreme case, the OEM may also decide to modify the systems in order to meet the criteria in the contract.

Under a performance based contract, the customer pays the OEM based on the performance. This means – in practice – that the earnings of the OEM are proportional to the performance of a system. That is, if a system is performing better (worse) than a specified target, the OEM is rewarded (penalized) for this. In the printing industry, companies such as Xerox and Océ use performance based contracts that reward OEMs if more pages are printed, because customers pay per printed page. On the other hand, OEMs pay penalties if certain availability criteria are not met.

The contracts motivate OEMs to reduce the total costs that are accrued over the life cycle; we call these costs *life cycle costs* (LCC). Life cycle costs can be considered from an OEM or customer perspective. An OEM distinguishes between design and production costs, while these costs are included in the purchase costs for a customer. If we take a customer perspective, the LCC are also known as total cost of ownership, see for instance Ellram (1994). In this thesis, we restrict our attention to OEMs and do not use the term total cost of ownership further.

OEMs that close service contracts with their customers are interested in lowering the LCC as they carry responsibility for the major part of the life cycle. The OEM's efforts for reducing the LCC start as early as the needs and design phase, when the system is developed. The decisions made during design vary from product architectural decisions, to detailed design decisions, to decisions for maintenance and operations. It is crucial for an OEM to make the right decisions at this stage since design decisions determine 70–85% of the LCC (Asiedu and Gu, 1998), and they have a major impact on the systems' performance. An example of a design decision that affects the LCC is the material quality that is used for the system: higher quality materials increase the reliability and thus lower the costs in the usage phase, because fewer failures occur. We have depicted the ability to influence the LCC, and the development of the LCC over the life cycle in Figure 1.2.

As soon as the design is fixed, the system is produced either by a company's production department or by a contract manufacturer. The production phase has a substantially lower ability to affect the LCC compared to the design phase, and the majority of the LCC has not yet been incurred after production. Various authors report that only 20–30% of the LCC have been incurred from design until production (Saranga and Kumar, 2006; Öner et al., 2007; van Dongen, 2011). This is also illustrated in Figure 1.2. Nevertheless, production deficits may still affect the LCC in later phases, for example production errors may induce poor performance in the usage phase and can even trigger costly system modifications.

Once the systems have been produced and shipped to the customer, it becomes

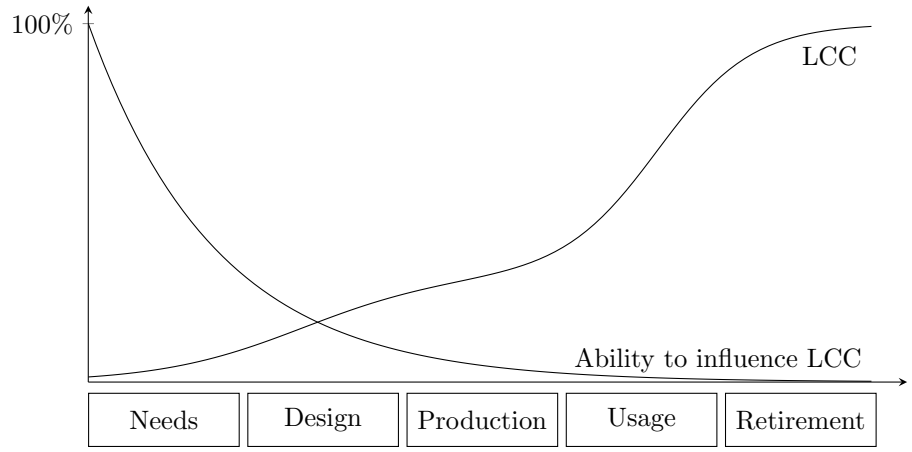


Figure 1.2: The development of life cycle costs, based on Norman (1990)

relatively hard to influence the LCC. However, if we can influence the LCC, it may have large consequences, because the majority of the LCC is incurred during the usage phase (Saranga and Kumar, 2006; Öner et al., 2007; van Dongen, 2011). The costs incurred in the usage phase are typically comprised of downtime costs, maintenance costs and costs of operations (Öner et al., 2007). The downtime costs are opportunity costs that are incurred when a system is not functioning, i.e., costs due to lost revenues, penalties (as stated in the contracts), idle resources, loss of customer goodwill etc. If we consider systems that require large financial investments, these downtime costs may be very considerable. For example, the downtime costs of a brokerage system for a large investment bank or brokerage company is approximately \$100,000–\$1,000,000 per hour (CNET News, 2001). In other settings, e.g. the semiconductor industry and baggage handling systems, the downtime costs are also substantial (Parent, 2000; Patterson, 2002) and they may constitute up to two thirds of the usage costs (Öner et al., 2007). To avoid these costs, companies perform maintenance activities to reduce the downtime of their systems. Maintenance may not only reduce the frequency of system failures, but it can also reduce the time that a system is down by quickly resolving the system failure. However, doing maintenance comes at a cost, and the maintenance costs over the life cycle can be significant. For instance, Öner et al. (2007) find in a case study at a large engineer–to–order company that the maintenance costs constitute roughly one third of the usage costs. This means that OEMs should be deeply concerned about the costs for downtime and maintenance in the usage phase.

Capital intensive systems are characterized by the fact that their usage phase is typically long – it commonly varies between 10 and 30 years. During these years, OEMs attempt to lower both the downtime and maintenance costs by enhancing system performance. This means that during the usage phase, the system is modified.

These modifications can lower the life cycle costs later in the life cycle, by improving productivity of the system, increasing the reliability or lowering the operating costs.

Finally, the system ages and reaches its retirement phase. During this phase, the system may be replaced by an other system, and the old system can be recycled, sold to a secondary market, or scrapped. The costs associated to this life cycle phase typically constitute a rather small part of the LCC.

1.1.1 Lowering life cycle costs

The total costs accrued over a life cycle are the result of a large number of decisions that are made by the OEM in various phases of the life cycle. Many of these decisions affect the performance and costs later in the life cycle. For instance, if we design a system such that it uses more reliable materials, the number of failures in the usage phase may decrease. Another example that affects the costs later in the life cycle is system modifications. A system is typically modified by replacing a part from the system with an improved part. Such an improved part may have a lower failure rate, enable higher system productivity and so on. Hence, system modifications influence the performance later in the life cycle, despite the fact that we remain in the usage phase.

It is clear that many OEM decisions – that are made at various points throughout the life cycle – have an effect later in the life cycle. In this thesis, we take the perspective of such an OEM.

1.2. Research problems

We identify three problems for which decisions determine (part of) the life cycle costs incurred later in the life cycle. These problems are discussed in Sections 1.2.1, 1.2.2, and 1.2.3. These sections sketch the problems and provide an insight why these problems are not trivially solved.

Most prototypical problems that affect the costs later in the life cycle can be found in the design phase, because this is the most upstream phase of the life cycle and 70–85% of the LCC are determined in this design phase (Asiedu and Gu, 1998); see also Figure 1.2. As a consequence, two of the three problems that we study are problems encountered in design. Furthermore, we also consider a problem in the usage phase, because this phase can be long (10–30 years for capital intensive systems) and it accounts for the largest part of the LCC.

In the design phase, we do not have any concrete and physical objects, which are existing in the usage phase. Therefore, we make a distinction between an abstract concept of an object and the physical counterpart. We call the abstract concept of an object a component, and the physical counterpart a part. For instance, an

engine during design is a component, while the produced engine with serial number 001246839 is a part.

1.2.1 Service part effects in commonality and reliability decisions

Design problems are highly complex, because they involve numerous interrelated decisions that have to be made. These decisions affect system features such as productivity, reliability and so forth. Moreover, the decisions made in the design phase determine the majority (70–85%) of the life cycle costs. Hence, it is crucial for an OEM that he makes the right design decision such that the life cycle costs are minimized. Another important aspect is that companies are compartmentalized (due to the organizational structure). As a consequence, designers tend to neglect operational (after–sales) aspects in their design decisions, despite the fact that these operational costs account for 70–80% of the life cycle costs (Saranga and Kumar, 2006; Öner et al., 2007; van Dongen, 2011). Therefore, we study how much an OEM may gain in terms of life cycle costs when he considers operational costs – such as costs for service parts provisioning and downtime – in the design decisions.

We restrict our attention to components of systems and we focus on two design decisions: commonality and reliability. Commonality means that the OEM decides to use a common component for multiple system types. On the other hand, if the OEM does not opt for commonality, he uses a dedicated component for each system type. For instance, if a car manufacturer considers two models and uses the same engine in both models, we say that the engine is common. Alternatively, if the car manufacturer uses a different engine for each model, we call each engine dedicated (it is dedicated to a model).

The other design decision that we consider in this problem is reliability. If the OEM has chosen commonality, he determines the reliability for the common component. Similarly, he has to determine the reliability for each of the dedicated components if the OEM selects dedicated components. The OEM's designers can achieve higher reliability by using higher quality materials, spending more time on conceptual and detailed design ideas etc. But all of these options typically increase costs in the design and production phases.

Both these design decisions are made in the design phase of the life cycle, but have major effects on the usage phase where the majority of the LCC is incurred. These effects can be found, particularly, in the area of after–sales services such as repairs and service part provisioning. If a part (common or dedicated) fails, the system cannot operate and a maintenance engineer is sent to the failure to fix it. If the maintenance engineer repairs the failed part on site, the system would be idling for a long period of time, which is very expensive. In order to prevent long downtimes of systems, OEMs typically keep service parts on stock. Then, if a part fails, a maintenance engineer

takes a working part from stock, swaps the failed part with the working one, and sends the failed part to an offline repair shop. Once the failed part is repaired it returns to stock. This strategy reduces the time and costs of idle systems, but it introduces costs of keeping service parts on stock.

If we use dedicated components, we must keep separate stocks for each of the components. Alternatively, if we use common components, only one stock of common service parts has to be kept and when we have a smaller number of stock piles, we can realize the same customer service levels with a lower total number of service parts on stock and thus lower costs (Baker et al., 1986; Hillier, 2000; Song and Zhao, 2009). This is one way that the commonality decision affects the costs that are incurred in the usage phase.

Reliability also influences the costs incurred later in the life cycle, affecting the number of failures observed in the usage phase. Since the OEM incurs a cost per failure (failed parts are repaired etc.), reliability has an immediate effect on the costs in the usage phase. Moreover, reliability also influences the stock levels of service parts. Higher (lower) reliability results in a lower (higher) the demand for service parts and thus lower (higher) optimal stock levels of service parts.

It is clear that our two considered design decisions determine the costs incurred later in the life cycle. In particular, both decisions have an effect on service part stocking and the costs associated with it. Therefore, there may exist a potential to lower the life cycle costs when the OEM considers service part stocking in its commonality and reliability decisions. But how large can such a reduction in life cycle costs be? Furthermore, how will the commonality and reliability decisions change if one considers service part stocking in the decision process?

1.2.2 Line Replaceable Units

In a later stage in the design phase of the life cycle, designers must decide the maintenance plan to use. Analogous to the previous problem, the OEM can strongly determine the costs that will be incurred later in the life cycle. The maintenance plan prescribes the maintenance actions that are executed during the usage phase of the life cycle. This means that the maintenance plan affects the downtime costs and the maintenance costs, which are the predominant cost factors in the usage phase (Öner et al., 2007). Thus, it is essential that the OEM carefully decides on the maintenance plan in order to minimize the life cycle costs.

A maintenance plan is typically determined for a given system design, making the system design an input into the maintenance plan. The maintenance plan could also be adjusted or even created later than the design phase. Therefore, we use terminology of parts rather than components for this problem. Furthermore, the maintenance plan prescribes which parts are maintained every fixed number of periods, which are continuously monitored, and which are replaced once they fail. It also prescribes so–

called Line Replaceable Units (LRUs). A LRU is a collection of parts that is replaced entirely from the system when one of the parts in the LRU has to be maintained. If we consider a racing bicycle used in stage races (e.g. Tour de France or Giro d'Italia), the rear tire, rear rim the spokes, the wheel hub[3], and the cassette are one LRU (rear wheel) and this LRU (rear wheel) is replaced if any of its parts fails.

The design of LRUs can have a big impact on the downtime costs and the maintenance costs. LRUs have the ability to decrease the time a maintenance engineer spends on replacing failed parts: if certain parts are combined together in a single LRU, it may be easy to remove and replace the failed LRU from the system. Whether it is easier to combine parts in a LRU depends on the system's design. Thus, there exists a potential to decrease system downtime. For instance, consider a racing bicycle that has a broken rear wheel spoke. Replacing the broken spoke itself is difficult and time consuming, while replacing the entire rear wheel (consisting of a tire, rim, spokes, wheel hub, and cassette) may be far easier and thus reduces the time spent on replacements. On the other hand, failed LRUs are replaced by new ones and these new ones need to be purchased (or failed ones repaired). However, when LRUs are small (contain fewer parts) they are typically cheaper to purchase (or repair) because they contain less value compared to larger LRUs (contain more parts). Hence, the costs for maintenance (due to purchase or repair) are also a direct consequence of the LRU's design. In our bicycle example, if we only purchase a new spoke, we incur lower purchase costs than ordering an entire rear wheel (consisting of the tire, rim, spokes, wheel hub, and cassette).

Thus if we design LRUs, we not only have to consider the design of the system, but also the downtime and maintenance costs in terms of replacement costs and purchase costs, respectively. This makes the task of designing LRUs intricate.

1.2.3 Implementation of system modifications

Once the systems are designed and produced, they are used for a certain amount of time. In the context of systems that require high financial investments, this usage period is long and may be of the order of magnitude of 10–30 years. Furthermore, up to 70–80% of the life cycle costs can originate from the usage phase (Saranga and Kumar, 2006; Öner et al., 2007; van Dongen, 2011). Hence, if the OEM can reduce the operational costs, even when he finds himself in the usage phase of a life cycle, this may be highly profitable.

One alternative frequently used by OEMs to lower the costs in the usage phase is to modify the used systems. System modifications are developed to enhance the performance of the system in terms of increased productivity, increased reliability, or reduced operating costs (e.g. lower power consumption). A system modification

[3] A wheel hub (in Dutch: *naaf*) is the central part of a wheel from which the spokes radiate to the rim.

means that a specific part in a system is replaced by an improved one, e.g. we replace an electric motor in a train by an improved one. A part can be on various levels in the system's bill of materials. For instance, we can consider an electric motor as a part of a train or the brushes in an electric motor as part of a train. In this problem, we consider parts that can be on any level in the system's bill of materials.

We study an OEM that has closed a service contract (e.g. a performance based contract) with its customers. Therefore, the OEM is rewarded (penalized) for better (worse) performance, and thus he is responsible for a number of capital intensive systems. These systems are installed in the field for a remaining lifetime (say 10–30 years), and each system consists of multiple parts. In this problem, we consider critical and repairable parts, and each part occurs once in a system. For instance, we consider a single part (positioning sensor) in a lithography system. As the OEM carries responsibility for the performance of its installed systems, he holds inventory for the parts that are installed in the systems, e.g. the OEM has a number of positioning sensors on stock in addition to the ones installed in field systems. At a certain moment, the OEM believes that the current parts (positioning sensor) suffer from a too high a failure rate, or that it is a bottleneck for the system's productivity. Therefore, he can develop a new and improved component with better performance. As a result, the OEM has to determine whether it is profitable to replace all current parts by new ones, and if it is he has to decide when to replace the current parts by new ones. The current parts are referred to as old parts in the remainder of the chapter.

The OEM can follow various strategies to implement the new parts. Given that a number of old parts are installed in field systems and that there exists a stock of old parts, an implementation strategy must determine how many new parts the OEM produces and how the new parts are implemented. Furthermore, the implementation strategy prescribes how many old parts from stock are salvaged, because these may become redundant. Implementing the new parts instantaneously induces an additional cost of visiting all installed systems and replacing the old parts with new parts, but it generates extra revenue because the OEM can immediately reap the benefits of the new parts. Furthermore, the OEM has to ensure that a sufficient number of new parts is produced for this strategy.

The OEM may also pursue a more conservative strategy wherein an old part is replaced by a new part once the old part has failed (or is undergoing maintenance), and all old parts from stock are immediately salvaged. This may reduce the number of new parts that is needed, because he gradually implements new parts. An even more extreme strategy is to not implement the new parts and stick to the old ones; then the OEM does not need to produce any new parts. In addition to all of this, the problem is further complicated by the fact that there are old parts (on stock) that need not to be salvaged because the OEM can choose to use these old stock parts instead of installing a new part.

As we see, there are many implementation strategies that the OEM can follow, but

pursuing the wrong strategy can be a costly endeavor. We may miss out on the performance gain of new parts, e.g. we are not able to reap productivity increases or reliability increases; or we could pursue a strategy that has unnecessary high costs for immediate implementation. Therefore, it is of utmost importance to select the best implementation strategy that considers the current costs of producing new parts along with the cost effects later in the usage phase of the life cycle.

1.3. Research objective

The three problems that we discussed in the preceding section are difficult problems to study without the use of decision support mechanisms. Intuitively it is hard to estimate, for instance, how large the benefit is of considering service parts in a commonality and reliability decision. Similarly, it is very challenging to see which LRU design results in low costs in the usage phase, taking into account the technical aspects of a system's design. Moreover, selecting the appropriate implementation strategy for new parts under the presence of old part stocks is not an easy task to perform, because it is not a priori clear which strategy is best.

In order to make the right decisions, there exists a need for advanced mathematical decision support models, which we provide in this thesis. We take the perspective of an OEM who sells and supports systems, and is interested in lowering (part of) the life cycle costs. Our objective is the following.

> *Develop mathematical decision support models to support the following three decisions:*
>
> *(i) Optimize commonality and reliability in the presence of service part stocks,*
>
> *(ii) Optimize the design of Line Replaceable Units such that the relevant (future) usage costs are minimized,*
>
> *(iii) Select the optimal implementation strategy for new parts, taking into account the future benefits of these new parts.*

1.4. Contributions of thesis

The contributions of this thesis are organized in accordance with the problems that we discussed in Section 1.2. The positioning of our work in the literature is covered in the corresponding chapters.

1.4.1 Service part effects in commonality and reliability decisions

In Chapter 2, we study the commonality and reliability (in terms of mean time between failure) decision both in the presence of service part stocks and in the absence of service part stocks. We consider the benefit of considering service parts for commonality and reliability decisions. If the OEM selects commonality it means that all systems will use the same common component, whereas no commonality (or dedicated components) means that each system uses a dedicated component. Furthermore, the OEM determines the reliability for each alternative (common or dedicated).

We study two approaches: one in which the OEM considers costs for production and repair, and where the OEM neglects service parts for the commonality and reliability decisions. In the second approach, the OEM considers production and repair costs, as well as holding costs and downtime costs because the OEM includes service parts in his commonality and reliability decisions. For each approach, we propose a model for a common component and another model for dedicated components. Depending on whether service parts are considered, each model optimizes the reliability level(s) and the service part stock level(s) (if applicable).

If the OEM considers service parts in his design decision, the optimization models of the common and dedicated components are intractable. We remark that we study settings in which systems require high financial investments and users heavily rely on the availability of these systems (CNET News, 2001; Patterson, 2002; Öner et al., 2010); thus the cost of system downtime is large. Consequently, we study two approximate models that are asymptotically equivalent as the cost of system downtime tends to infinity. When the OEM neglects service parts from its design decision, the optimization models are tractable.

We prove convexity of the cost functions for both approaches, and this enables straightforward optimization of the reliability. We compare the optimal reliability levels between the two approaches, and we observe that considering service parts can result in optimal reliability that are 27% higher than the optimal reliability levels obtained when neglecting service parts; on average this difference is 10%. Such differences are large and affect operations in the usage phase of the life cycle, in particular. As the costs of usage can account for the majority of the life cycle costs, the OEM should be motivated to consider service parts in the reliability decision. Furthermore, we characterize a switching curve for each approach (considering service parts or not) that determines whether the OEM selects commonality or not. The switching curve also provides an insight in when commonality is *not* a good idea. We analytically and numerically compare the switching curves between both approaches. We find that commonality is strictly favored if we consider service parts in the design decision, and this more favorable attitude towards commonality persists even if the unit cost of a common component increases by as much as 9.5%. This means that

service part considerations should be considered if a good commonality is to be made: the OEM can invest significantly more in the common component (it can become more expensive) and still obtain lower life cycle costs if he considers service parts in the design decisions. Finally, we show that including service parts in commonality and reliability decisions may lead to much lower life cycle costs, being as much as 10% lower. Hence, neglecting service parts can be very costly and may even be detrimental to the OEM's profitability. Hence, the OEM is urged to consider service parts in the commonality and reliability decision, if he wants to minimize the life cycle costs.

1.4.2 Line Replaceable Units

Chapters 3 and 4 study the design of Line Replaceable Units (LRUs). Chapter 4 is an extension to Chapter 3, and therefore we will first discuss the contribution of Chapter 3 and subsequently we address the additional contribution of Chapter 4.

If designers study the problem of designing LRUs, they find themselves later in the design phase. As a consequence, the system's design is typically input to the problem of designing LRUs. Therefore, we start Chapter 3 by extensively discussing how we represent a system's design for maintenance applications. In particular, this means that we discuss an approach to model a system in terms of parts and the connections between them. If we consider the rear wheel of a bicycle, it consists of multiple parts such as a tire, rim, spokes, a wheel hub, and the cassette. We represent each part as a vertex and we connect two vertices when two parts are connected to each other. In the bicycle example, this means that the rim is connected to each of the spokes, and each of the spokes is also connected to the wheel hub. In addition to the system representation in terms of parts and their connections, we also present a technique that enables us to include a disassembly sequence in our model, i.e., we incorporate a sequence of connections that needs to be broken prior to breaking a certain connection. This approach makes our model particularly applicable for maintenance applications, because disassembly sequences frequently exist in such settings. For example, if we want to remove a spoke of a bicycle's rear wheel, it has to be disconnected from the rim. However, we first have to remove the tire from the rim to disconnect the spoke from the rim.

This system representation can be used as a (visual) aid to enhance internal communication at the OEM; for instance between the design department and the operations department. Given the system representation for maintenance in place, we make an important assumption in Chapter 3 that each part belongs to exactly one LRU. Under this assumption, we formulate a model called LRU Design that trades off the costs for replacing LRUs and the costs of purchasing (or repairing) LRUs. We present the most natural formulation of LRU Design: a binary non–linear program. Subsequently, we linearize this program to a binary linear program (BLP). We also formulate LRU Design as a set partitioning problem. Typically, one would solve such a set partitioning problem by using branch–and–price algorithms. We prove

that branching is unnecessary and the set partitioning formulation can be solved by pure pricing algorithms (and an optimal integer solution is obtained). Our numerical study illustrates that set partitioning is an efficient formulation and suitable for large instances, while BLP is not. LRU Design – and in particular the set partitioning formulation – can be used as a feedback mechanism for the OEM's design department. The engineers can quickly assess various design alternatives and their effects on the optimal LRU design and the corresponding (after–sales) costs. This could aid the OEM in making better decisions that reduce the life cycle costs. Furthermore, we find that the OEM should urge his designers to avoid strongly connected parts, and to avoid intense disassembly sequences because these induce high operational costs.

In Chapter 4 we extend the model of Chapter 3 by relaxing the assumption that a part belongs to exactly one LRU. This means that we allow replacement of a LRU containing a certain part, even when the failure of this part does not trigger the replacement of the LRU. For instance, consider a student that has a city bike. If one of the spokes in the rear wheel breaks, the student replaces only the broken spoke because she finds it too expensive – relative to the time she needs for the replacement – to purchase an entirely new rear wheel (except the tire). However, if the rim breaks, the student has to spend a lot of time and effort when she only replaces the rim itself. Therefore, she decides to purchase an entirely new rear wheel (except the tire) if the rim breaks. This means that all spokes are also replaced. Hence, if a spoke fails only the broken spoke is replaced, whereas a rim failure induces the replacement of the entire wheel including all spokes. Thus, we say that a spoke belongs to more than one LRU, and this phenomenon is studied in Chapter 4. As a consequence, we conceptualize a LRU slightly different from Chapter 3.

We use this re–conceptualized LRU to present a model C–LRU Design that trades off replacement and purchase (or repair) costs. We observe that C–LRU Design is separable in the number of parts. For each part we present a binary linear program, and numerically we show that this formulation is efficient, even for large instances. Moreover, we study the differences in computation time, minimum costs, and the number of LRUs used between LRU Design and C–LRU Design. The computation times are so low that C–LRU Design is very practical and it can be used as a feedback mechanism to the OEM's design department, thereby lowering the life cycle costs. Moreover, we find the that strongly connected parts and intense disassembly sequences should be avoided by designers.

1.4.3 Implementation of system modifications

Chapter 5 studies the problem of modifying the systems that are currently installed in the field. We restrict our attention to replacing old parts by new parts under the presence of an old parts on stock, as discussed in Section 1.2. The OEM focuses on four different implementation strategies under a finite horizon:

- Instant Invest: The OEM produces all new parts before the start of the horizon and decides whether to replace a failed part by a new or old one upon each failure. He repairs all new failed parts and old parts are salvaged. Salvaging of old parts occurs during and at the end of the horizon, while salvaging of new parts only occurs at the end of the horizon.
- Phased Invest: The same strategy as Instant Invest, except for the fact that the OEM produces some parts at the start of the horizon and some parts arrive during the horizon.
- Stay Put: The OEM does not produce any new parts, repairs the old parts if they fail, and salvages old parts at the end of the horizon.
- Rapid Upgrade: The OEM produces new parts and directly replaces all old parts by new parts. He salvages the old parts immediately (at the start of the horizon). The OEM repairs the new parts once they fail, and he salvages new parts at the end of the horizon.

Instant Invest and Phased Invest are advanced implementation strategies that are further developed in Chapter 5. The other two strategies – Stay Put and Rapid Upgrade – serve as benchmark strategies and are fairly easy formulations.

We start Chapter 5 by assuming a fixed production quantity of new parts, and we formulate Instant Invest as a finite horizon, discounted, discrete time Markov decision process that maximizes profit. We numerically show, for practical instances, that a failed part is replaced by a new one (if new parts are available), otherwise we use an old part from stock. If an old part is also not available, then the failed part will not be replaced. However, this behavior does not hold in general, as we observe in Chapter 5. Subsequently, we discuss how to find the optimal production quantity of new parts under Instant Invest. In the next part of Chapter 5, we extend Instant Instant to Phased Invest, where again we start with the assumption that the production quantities of new parts are given. Furthermore, we assume that the second production order is planned prior to the horizon and is thus known. Then, we also find that we replace a failed part by a new one (if available) in practical instances, but this behavior does not necessarily occur for an arbitrary instance. Lastly, we discuss how to determine near–optimal production quantities and the arrival time of the second production order.

In numerical experiments, we find that the OEM should not replace the old parts by new ones (Stay Put is optimal), if the new component is not (or only slightly) better than the old one. On the other hand, if the new component is significantly better than the old component, the OEM should replace the old parts by new ones as soon as possible (Rapid Upgrade is optimal). If the new component is somewhat better in terms of generated revenue, the OEM should gradually replace the old parts by new ones (upon the failure of an old part). That is, we observe that the advanced strategies Instant Invest and Phased Invest are preferred. In particular, Phased Invest

generates strictly more profit than Instant Invest and Phased Invest is optimal in such cases.

1.5. Thesis outline

This thesis presents and analyzes the three different problems from Section 1.2 dispersed over four chapters. Each of the chapters is set up in such a way that it can be read individually, except for Chapters 3 and 4 because the latter is an extension of the former. Figure 1.3 depicts the structure of this thesis schematically. Each problem uses a different methodology: Chapter 2 mainly uses asymptotic analysis and general function analysis; Chapters 3 and 4 use graph theory and combinatorial optimization theory such as binary programming and column generation (only for Chapter 3); and Chapter 5 uses Markov decision theory as the main methodology.

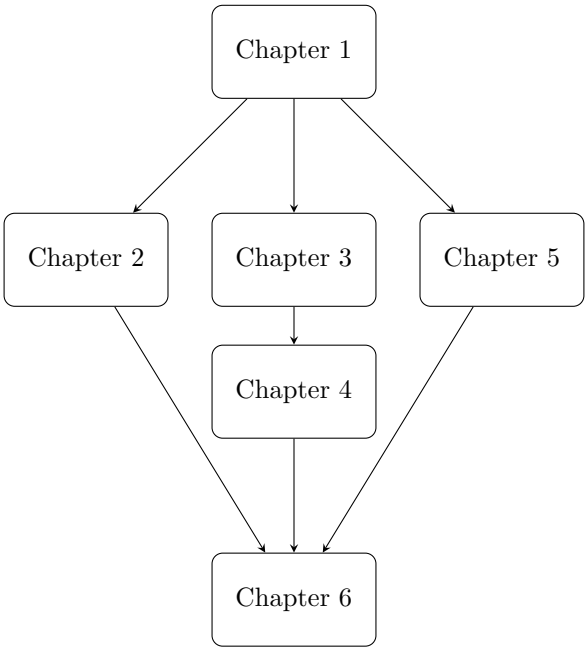


Figure 1.3: Thesis structure

2

Service part effects for commonality and reliability decisions

2.1. Introduction

In this chapter, we study an OEM that has closed a service contract (e.g. a performance based contract or a full service contract) with his customers. As a consequence, he is responsible for the entire life cycle of systems, and is therefore primarily interested in minimizing the total life cycle costs. Examples of such OEMs include Pratt & Whitney (aviation industry), ASML (semiconductor industry), and Océ (industrial printing industry). The life cycle costs (LCC) increase as OEMs offer a higher variety of systems. In an attempt to alleviate this burden, OEMs use common components in multiple different systems of their product portfolio. For example, identical rotor blades are used in multiple different aerospace engines or the same positioning sensors are used in various lithography systems. The main motivation for a variety of systems with commonality comes from a marketing and production perspective, as it enables a firm to offer many different systems with a relatively small increase in the production costs. However, a commonality decision has more effects, because it influences after–sales services and the effects may be large. In particular, commonality enables service parts pooling which reduces costs. This cost benefit of common service parts – despite the potentially increased production costs – is particularly large in industries that use capital intensive systems. Therefore, one would expect that service part aspects are considered for the commonality decision. However, we frankly observe that design departments tend to omit service part considerations in the commonality decision due to the organizational structure in a

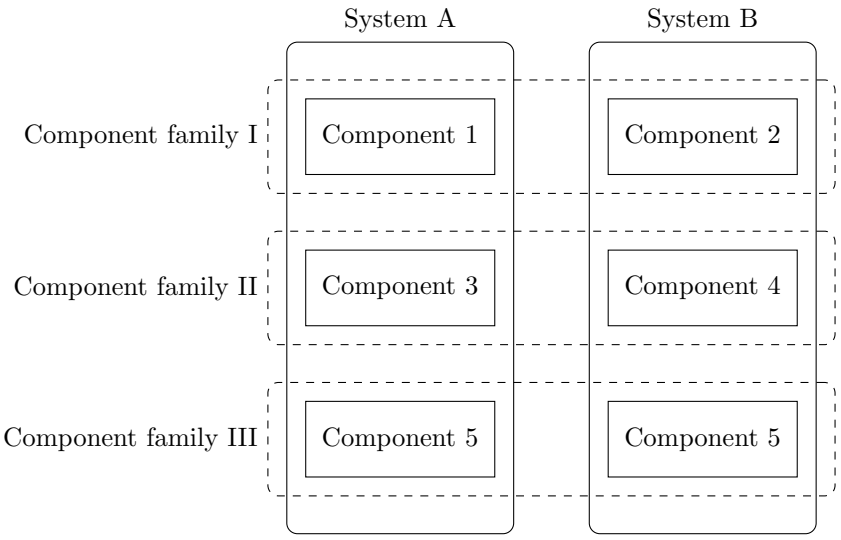


Figure 2.1: A schematic representation of our concepts

company: the design department carries no responsibility for after–sales performance, while the after–sales department may suffer from decisions made in the design department. Thus, design departments are typically myopic in the sense that they do not consider service parts aspects in their commonality decision.

We conceptualize commonality as follows. A system consists of components, and we say that components – from different systems – belong to the same *component family* when they fulfill the same functionality, but are not necessarily identical (Meyer and Lehnerd, 1997). Therefore, the OEM can decide per component family whether to use a single common component for all systems or to use a dedicated component per system, as illustrated in Figure 2.1. We do not consider partial commonality, where a common component is used for a subset of the systems. A component family may correspond to rotor blades in the case of aerospace engines, to positioning sensors in lithography systems, or to electric motors in MRI scanners.

Next to the commonality decision, the OEM can reduce the LCC by optimizing the reliability of a component. We capture reliability in terms of the mean time between failures. A component's reliability largely affects after–sales performance because reliability determines how often a component will fail. Furthermore, reliability may serve as a substitute for service parts (Kim et al., 2017). This means that an increased (reduced) reliability can reduce (increase) the investment in service parts. Therefore, there exists a trade–off between the cost of reliability and the cost of service part inventory holding. Hence, an OEM that considers service parts in his design decision does not only gain on the commonality aspect, but can also exploit the substitution effect that reliability and service part inventories have. Considering service parts in

design decisions appears to be a smart idea, particularly when we consider the fact that after–sales services, including service parts provisioning, constitute up to 70–80% of the LCC (Öner et al., 2007).

Therefore, we study how much an OEM gains when he considers service parts in the design decision in terms of commonality and reliability. Our objective is to answer the following main research questions for a single component family: (RQ1) What is the difference in the optimal reliability decision when we consider service parts? (RQ2) Is commonality more or less attractive when we consider the consequences for service part inventories? (RQ3) How much of the LCC can we save when we consider service parts in the design decision in terms of commonality and reliability?

We consider two approaches: one in which the OEM considers service parts, and one in which the OEM excludes service parts from the commonality and reliability decision. We refer to these approaches as the *anticipating approach* and the *non–anticipating approach*, respectively. If the OEM considers service parts in his decision making (the anticipating approach), his decision problem is more intricate. The OEM determines not only commonality, but he also makes a conscious decision on the trade–off between reliability and service part stock levels. Therefore, the OEM that uses the anticipating approach minimizes the LCC by optimizing a decision triad: (1) component commonality or dedicated components, (2) the reliability for the components of each alternative, and (3) the service part stock levels for the components of each alternative. In case the OEM pursues the non–anticipating approach, he minimizes a different LCC function by optimizing a decision dyad consisting of (1) and (2). Furthermore, he does not exploit the beneficial effect that commonality has on service parts pooling and the substitution effect between reliability and service part stock levels.

For each approach, we develop two LCC models; one for the common component and one for the dedicated components. The goal of the OEM is to select the alternative – common or dedicated – with corresponding optimal reliabilities and service part stock levels (if considered) such that the total LCC is minimized.

Based on the models for both approaches, we make the following contributions. First (1), we show that the problem formulations for the anticipating approach are intractable. Therefore, we propose approximate problem formulations, and we prove that these approximations are asymptotically equivalent as the cost of system downtime tends to infinity. Furthermore, the approximate problem formulations enable us to efficiently optimize the reliability levels for the common and dedicated components. The problem formulations for the non–anticipating approach are simpler than the formulations of the anticipating approach, and as a consequence the non–anticipating formulations do not suffer from intractability issues. Secondly, (2) we numerically study the differences in the optimal reliability decisions under both approaches (RQ1). We conclude that neglecting service parts for the reliability decision can be detrimental as we observe instances wherein the optimal reliability level is 27% higher under the anticipating approach, and on average it is 10% higher.

Thirdly, (3) we exactly characterize the switching curve – for each approach – that determines whether commonality yields lower LCC than dedicated components. This switching curve is based on the costs of the common component relative to the costs of the dedicated components, and on the installed bases of different systems. The switching curve also provides an insight in when commonality is *not* a good idea. Fourthly, (4) we analytically compare the switching curves for both approaches (RQ2) and conclude that the anticipating approach strictly favors commonality compared to the non–anticipating approach. Furthermore, we observe in a numerical study that we obtain different decisions under both approaches even if the unit costs of a common component increases by as much as 9.59%. Finally, (4) we numerically show that an OEM reduces the relevant LCC by as much as 10% if he pursues the anticipating approach over the non–anticipating approach (RQ3).

The organization of this chapter is as follows. In Section 2.2, we discuss related literature. In Section 2.3, we present a commonality model and a dedicated components model for each approach (anticipating and non–anticipating). We continue in Section 2.4 by optimizing the reliability and the service parts stock levels (if applicable) for the common and dedicated component models, under the anticipating and the non–anticipating approach. Furthermore, we compare the differences in the optimal reliability levels between both approaches (RQ1) in Section 2.4. Given the optimal reliability and service parts stock levels (if applicable), we study the commonality decision under both approaches in Section 2.5, i.e., for each approach we select the model (common or dedicated) with the lowest life cycle costs. Moreover, we compare the difference in the commonality decision (RQ2) in this section. In Section 2.6, we study the difference in LCC (RQ3) between the anticipating and the non–anticipating approach. Finally, we conclude this chapter in Section 2.7.

2.2. Literature

Our work focuses on the interaction between three literature streams: reliability optimization, after–sales services, and commonality. The literature on reliability optimization is typically studied from an engineering perspective. Design variables are chosen such that a specified cost function is minimized and reliability constraints are satisfied, or the reliability is maximized under specified cost constraints (Royset et al., 2001; Zou and Mahadevan, 2006). Reliability optimization is also studied in the broader context of the warranty literature, e.g. Huang et al. (2007) and references therein. Work in this stream typically aims to optimize the reliability of a component such that the costs that are accrued throughout the warranty period are minimized. The second related literature stream focuses on after–sales services, particularly on service parts planning problems in several supply chain structures; see e.g. Sherbrooke (2004), Muckstadt (2005), and van Houtum and Kranenburg (2015). The third related literature area considers component commonality. This topic is studied from numerous different perspectives, e.g. marketing (Desai et al., 2001), new product

development (Muffatto and Roveda, 2000), and engineering (Fellini et al., 2004). We restrict our commonality review to an operations management perspective that focuses on cost minimization, and we refer to Labro (2004) for a more elaborate review. There exists a variety of works that consider stylized Assemble–To–Order (ATO) models and focus on the inventory implications of commonality; see for example Baker et al. (1986), Hillier (2000), van Mieghem (2004), and Song and Zhao (2009). Authors generally conclude that commonality allows for inventory pooling, which reduces the costs. van Mieghem (2004) notes, however, that a common component can be more expensive than the dedicated components, and thus the cost reduction due to inventory pooling may be offset by the extra cost of a common component. Hence, he presents a condition for the adoption of commonality. In this chapter, we also find such a condition. In addition to ATO research, various researchers have modeled component commonality in a broader context by considering other factors than inventories as well. Typically, these authors take a combinatorial approach by studying large mathematical programming models, see e.g. Gupta and Krishnan (1999) and Thonemann and Brandeau (2000). These works focus on deriving efficient solution procedures.

This chapter is closely related to research that focuses on the interaction between two of these three literature streams (commonality and after–sales services). Stylized inventory models with component commonality have mainly been studied for ATO systems. In an ATO setting, a product demand is satisfied if *all* components are on stock (coupled demand). This differs from an after–sales services setting, in which demand typically occurs for each of the individual components. Kranenburg and van Houtum (2007) take such an after–sales services perspective and focus on the service parts inventory implications of component commonality. They only model the costs incurred after production of the component, i.e., reliability and commonality decisions are neglected, but downtimes are included. Thonemann and Brandeau (2000) explicitly model the commonality decision and consider the service parts aspect of after–sales services, but exclude downtimes and repairs. Their problem is combinatorial and the authors determine which features one or more common components should have based on component requirements. Both papers, Thonemann and Brandeau (2000) and Kranenburg and van Houtum (2007), do not consider any reliability decision and focus on developing efficient solution techniques.

Another closely related literature stream studies reliability in combination with after–sales services. Research in this stream has not yet considered commonality. Huang et al. (2007) propose a profit maximization model that optimizes the reliability and considers sales revenues, production costs, and repair costs. They take a life cycle approach, but do not consider costs related to service parts. Kim et al. (2017)[1] present a LCC minimization model that jointly optimizes reliability and service part stock levels. The LCC are comprised of design, production, service parts storage, and backorder costs. Downtime and repair costs are not included. Kim et al. (2017)

[1]The work by Kim et al. (2017) has been first published as a working paper, see Kim et al. (2007b).

find that reliability and service parts are substitutes for each other, and they derive analytical insights for different types of service contracts through the use of a game theoretical analysis. We remark that Kim et al. (2007a) differs from this stream, because the authors do consider cost reduction efforts and service part inventories, but do not focus on reliability improvements that affect demand intensity. By contrast to Kim et al. (2017), Öner et al. (2010) do not take a game theoretic perspective, but focus on the after–sales services aspects of the LCC minimization problem. The authors extend the cost function from Kim et al. (2017) by also incorporating repairs and downtimes, and they assume that demand during a stockout is satisfied via an emergency procedure. Öner et al. (2010) develop an efficient algorithm, and find that substantial cost savings can be realized by simultaneously optimizing reliability and the service part stock levels.

Our work uses similar modeling as Kim et al. (2017) and Öner et al. (2010), but we also include the commonality dimension combined with the explicit inclusion of three after–sales services aspects: service parts, repairs and downtimes. We are the first to combine after–sales services, with commonality and reliability optimization into a single model. In order to analyze our models, we build on asymptotic techniques similar to Huh et al. (2009) and Bijvank et al. (2014). Moreover, we provide managerial insights via analysis instead of focusing on an efficient solution procedure. We provide a comparison between our work and the most related research in Table 2.1.

Paper	Commonality	Reliability	Downtime	Repairs	Service parts	Life cycle costs	Asymptotic analysis
Kranenburg and van Houtum (2007)	x		x	x	x		
Thonemann and Brandeau (2000)	x				x		
Huang et al. (2007)		x		x		x	
Kim et al. (2017)		x			x	x	
Öner et al. (2010)		x	x	x	x	x	
Huh et al. (2009)							x
This chapter	x	x	x	x	x	x	x

Table 2.1: Comparison of most related papers

2.3. Models

In this section, we first present the model and optimization problems of the anticipating approach (Section 2.3.1). Subsequently, we discuss the model and optimization problems for the non–anticipating approach (Section 2.3.2).

2.3.1 Anticipating approach

The OEM offers her customers various systems $i \in J$, where J is to the set of different systems. Furthermore, the OEM expects to sell $N_i > 0$ units of each system $i \in J$ at time $t = 0$ with a supplementary service contract. Due to such a contract the OEM is penalized worse performance. The service contract states that the OEM will service the N_i units of each system $i \in J$ for a finite lifetime $T > 0$. We assume that this time T is equal for all systems, and it is typically 10–30 years for capital intensive systems. Given the component structure for her systems (see Figure 2.1), the OEM has to determine for each component family whether to opt for a common component or dedicated components. We focus on a single component family, which is critical for system functioning and repairable upon failure. Furthermore, we assume that exactly one component of the family occurs in a system $i \in J$, e.g. one rotor blade assembly occurs in an aerospace engine or only one positioning sensor occurs in a lithography system. As a consequence, the sales quantity N_i of system i is equal to the number of parts of a dedicated component – for system i – that is installed in the field at time $t = 0$. This assumption can be generalized easily: if a component occurs x times in a system and each component operates independently, then we have xN_i parts of component i installed in the field at time $t = 0$.

Moreover, a system's identifier $i \in J$ is equivalent to a dedicated component's identifier $i \in J$, and thus the set J is equivalent to the set of the dedicated components. Note that each part will be serviced for a finite lifetime T. We refer to the parts of component i that are installed in the field, N_i, as the installed base of component i. In the remainder, we will use the terminology on the component level for the set J (the set of dedicated components) and for each element $i \in J$ (a dedicated component). Next to the notation for the dedicated components, we denote the common component by q with $N_q = \sum_{i \in J} N_i$, and introduce the set $I = J \cup \{q\}$.

At time $t = 0$, the OEM decides on the reliability level $\tau_i > 0$ in terms of the Mean Time Between Failures (MTBF) and on the turnaround stock level s_i of service parts. We use the term reliability in the remainder instead of MTBF. At time $t = 0$, $N_i + s_i$ parts of component i are produced with reliability τ_i. When choosing a higher reliability level, the unit production cost also increases, e.g. higher quality materials are used or additional production steps are needed. Moreover, the higher the reliability level, the higher the costs to increase the reliability level by one unit. This implies that the unit production cost is convex and increasing in the reliability level

τ_i, see also Mettas (2000) and Öner et al. (2010). Hence, we introduce the function $c : (0, \overline{\tau}) \to \mathbb{R}_+$, where $\mathbb{R}_+$ denotes the set of positive real numbers, i.e., $\mathbb{R}_+ = \{x \in \mathbb{R} \,|\, x > 0\}$. Furthermore, c is a twice differentiable, convex, and strictly increasing function with $\overline{\tau} \in \mathbb{R}_+ \cup \{\infty\}$, $0 \leq \lim_{\tau \downarrow 0} c(\tau) < \infty$, $0 \leq \lim_{\tau \downarrow 0} \frac{dc(\tau)}{d\tau} < \infty$, and $\lim_{\tau \to \overline{\tau}} c(\tau) = \infty$. We remark that the function c is identical for all components $i \in I$. However, the components $i \in I$ may be different: there may exist a low end, a medium end, and a high end component. For instance, a low end positioning sensor of a lithography system has a low resolution, while a high end sensor has a high resolution. Increasing the reliability of a higher end component is typically expensive, because such components are typically more advanced (technology wise). As a consequence, it is more complex and thus more expensive to improve its reliability. We model this by multiplying the convex function c by a so–called relative unit cost factor $\beta_i > 0$. This β_i enables differentiation between the various components $i \in I$ such that it is more expensive to increase the reliability for higher end components. Hence, the unit production cost for a component $i \in I$ is given by $\beta_i c(\tau_i)$.

After the N_i parts of component i have been installed in the field at $t = 0$, they operate independently with the same reliability τ_i. During operation, the parts fail and each failure triggers a demand at the stockpoint. We denote the demand during $[0, t]$ by $D_i(t, N_i, \tau_i)$ for any $t \geq 0$, and we assume that this demand process has independent and stationary increments. We also assume that $D_i(t, N_i, \tau_i)$ is normally distributed with mean $\mathbb{E}[D_i(t, N_i, \tau_i)] = \frac{N_i t}{\tau_i}$ and standard deviation $\sqrt{\alpha N_i t / \tau_i}$, where the constant $\alpha > 0$ is the variance to mean ratio. Such a normal distributed demand process enables us to obtain a closed form expression for the LCC under an optimal stock level. This makes further analysis tractable. Furthermore, we assume that the number of operating parts N_i remains constant, even when a part of the installed base fails. We make this assumption, because failed systems are down for small amounts of time in practical settings, and it is a common assumption in the literature; see Muckstadt (2005) and van Houtum and Kranenburg (2015). Our proposed demand process can approximate a (more conventional) Poisson process when $\mathbb{E}[D_i(t, N_i, \tau_i)]$ is sufficiently large and $\alpha = 1$, similar to Kim et al. (2017). However, if we were to employ a Poisson demand process, the analytical results could not all be established. We discuss the implications of Poisson distributed service parts demand in Appendix 2.B.

If a part fails, a service part is taken from stock (if possible) and it replaces the failed part. The failed part is sent to a repair shop with ample capacity, where it takes $L > 0$ time units to repair the failed part. After repair, the part is forwarded to the service part stockpoint, see Figure 2.2. Note that this stockpoint corresponds to a stockpoint operating under a policy with base–stock level s_i and a leadtime L.

Furthermore, if a part fails and there is no service part available at the stockpoint, the replacement of the failed part in the field cannot occur. Consequently, the system in which the failed part was installed cannot operate until a new service part is available again, i.e., we have a backorder. The OEM incurs a penalty $b > 0$ per unit time that

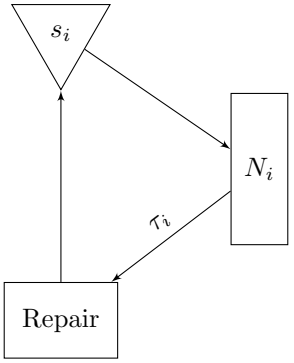


Figure 2.2: The failure and repair process for the parts of component $i \in I$

the system cannot operate due to service part unavailability. This b is the penalty specified in the service contract for each time unit that a system is more down than the specified target. We take the perspective that each time unit of downtime costs the OEM b relative to realizing 100% uptime.

Failed parts are repaired, and the costs per repair are related to the unit production cost of a component $\beta_i c(\tau_i)$. More expensive components (in terms of the unit production costs) are also more expensive for repair, because more expensive materials may have been used or repairing the part requires extra steps. Hence, we assume that the costs per repair are a linear scaling $r > 0$ of the unit costs, i.e., $r\beta_i c(\tau_i)$. This $r\beta_i c(\tau_i)$ includes all costs incurred for one repair such as material and labor costs. Then, the average number of repairs can be derived from our demand process and the total expected repair costs are given by: $r\beta_i c(\tau_i)\mathbb{E}[D_i(T, N_i, \tau_i)] = r\beta_i c(\tau_i)\frac{N_i T}{\tau_i}$.

The OEM owns all s_i turnaround service parts during $(0, T]$. Therefore, the OEM pays storage costs for all turnaround service parts, either in repair or in stock. The per time unit storage cost for one turnaround service part is a fraction $h \in (0, 1)$ of its unit cost, i.e., $h\beta_i c(\tau_i)$. The parameter h includes all per time unit costs for a single turnaround part, such as warehousing and insurance costs. The total service parts storage costs over $(0, T]$ are given by $hs_i T\beta_i c(\tau_i)$.

We have explained the dynamics for a given component $i \in I$ in the foregoing. These dynamics are identical for all dedicated components $i \in J$, and also for the common component q. Therefore, we formulate a general LCC function for component $i \in I$:

$$\tilde{\pi}(\tau_i, s_i, N_i, \beta_i) = \beta_i c(\tau_i)(N_i + s_i) + hs_i T\beta_i c(\tau_i) + r\beta_i c(\tau_i)\frac{N_i T}{\tau_i} + bT\mathbb{E}[(D_i(L, N_i, \tau_i) - s_i)^+]. \tag{2.1}$$

We do not consider an index i for the LCC function $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$, because each component $i \in I$ is fully characterized by its reliability level τ_i, turnaround stock level s_i, installed base size N_i, and its relative unit cost factor β_i. Hence, we

propose a general parametrized LCC function, which we can later analyze for arbitrary component types more easily. Furthermore, note that our model allows common components that are more expensive than each dedicated component, i.e., $\beta_q \geq \beta_i$ for all $i \in J$, as argued by van Mieghem (2004); but it can also capture the opposite, i.e., $\beta_q < \beta_i$ for one or more $i \in J$, as discussed by Krishnan and Gupta (2001). Furthermore, we can set $\beta_i = 1$ for one particular component $i \in I$ without loss of generality. To see this, define $\widehat{\beta}_i = \gamma\beta_i$, $\widehat{c}(\tau_i) = c(\tau_i)/\gamma$, with $\gamma > 0$ a constant, and set γ such that $\widehat{\beta}_i = 1$.

Now, the OEM of the systems seeks to minimize

$$(CP) \quad \min_{\tau_q \in \mathbb{R}_+, s_q \in \mathbb{R}} \left\{ \tilde{\pi}(\tau_q, s_q, N_q, \beta_q) \right\}, \text{ and}$$

$$(DP) \quad \min_{\boldsymbol{\tau} \in \mathbb{R}_+^{|J|}, \boldsymbol{s} \in \mathbb{R}^{|J|}} \left\{ \sum_{i \in J} \tilde{\pi}(\tau_i, s_i, N_i, \beta_i) \right\},$$

with $\boldsymbol{\tau}$ and $\boldsymbol{s}$ denoting the vector of τ_i and s_i, $i \in J$, respectively. The former model (CP) considers the common component, whereas (DP) considers the dedicated components. After solving each of the models, the OEM selects the alternative with minimum costs. Furthermore, we note that (DP) is separable in the dedicated components $i \in J$.

2.3.2 Non–anticipating approach

Next, we consider the problem for the non–anticipating approach, which omits service part considerations from the design decision. We use the same model description and notation as Section 2.3.1 and we define the LCC function $\hat{\pi}(\tau_i, N_i, \beta_i)$ for the non–anticipating approach. The derivation for this expression is identical to the derivation of $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$, and thus we do not elaborate on this. We have

$$\hat{\pi}(\tau_i, N_i, \beta_i) = \beta_i c(\tau_i) N_i + r\beta_i c(\tau_i) \frac{N_i T}{\tau_i},$$

because the service part stock levels are not considered. Subsequently, we derive the optimization models for the common component and for the dedicated components analogously to (CP) and (DP). Therefore, we obtain

$$(\widehat{CP}) \quad \min_{\tau_q \in \mathbb{R}_+} \left\{ \hat{\pi}(\tau_q, N_q, \beta_q) \right\}, \text{ and}$$

$$(\widehat{DP}) \quad \min_{\boldsymbol{\tau} \in \mathbb{R}_+^{|J|}} \left\{ \sum_{i \in J} \hat{\pi}(\tau_i, N_i, \beta_i) \right\},$$

where $(\widehat{DP})$ is separable in the dedicated components $i \in J$. The OEM solves $(\widehat{CP})$ and $(\widehat{DP})$ and he selects the alternative with the lowest LCC.

2.4. Optimal reliability and stock levels

We study the optimal reliability levels and optimal stock levels of the models (CP) and (DP) in Section 2.4.1. Subsequently, we study the optimal the reliability levels for $(\widehat{CP})$ and $(\widehat{DP})$ in Section 2.4.2.

2.4.1 Anticipating approach

We derive an expression for the optimal service part stock level $s_i^*(\tau_i)$ for component $i \in I$ and given reliability τ_i. Once we insert this $s_i^*(\tau_i)$ into (CP) and in (DP), we conclude that the problems are hard to optimize, because there exists a complex embedding of $c(\tau)$ in the cost functions $\tilde{\pi}(\tau_q, s_q^*(\tau_q), N_q, \beta_q)$ and $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$. Therefore, we propose an approximate total LCC function for the common and dedicated model, and these approximate total LCC functions are asymptotically equivalent to $\tilde{\pi}(\tau_q, s_q^*(\tau_q), N_q, \beta_q)$ and $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$, respectively, as the cost of system downtime b tends to infinity. The main benefit of these approximate total LCC functions is that it allows for straightforward optimization. Another major benefit of studying these approximate functions is that we can characterize a switching curve for commonality later in the chapter.

Before we start the optimization of (CP) and (DP), we state a mild assumption that we use for the anticipating approach throughout this entire chapter.

Assumption 2.1 For each component $i \in I$, we have $bT > 2\beta_i c(\tau_i)(1 + hT)$.

The interpretation of Assumption 2.1 is that having a system down over the horizon $(0, T]$ is more than twice as expensive as producing a new part and keeping this part on stock throughout $(0, T]$. Such an assumption is typically satisfied in practice, because downtime of capital goods is very expensive. Moreover, Assumption 2.1 constrains the feasible reliability levels τ_i such that the optimal turnaround stock level of component $i \in I$ is strictly positive. Such strictly positive turnaround stock level can only be guaranteed if $bT > 2\beta_i c(\tau_i)(1 + hT)$ because only then $\Phi^{-1}\left((bT - \beta_i c(\tau_i)(1 + hT))/(bT)\right) > 0$, see Lemma 2.1. Next, we determine the optimal turnaround stock level for a given reliability level by the following result.

Lemma 2.1 *For each component $i \in I$ and $\tau_i \in (0, \overline{\tau}]$, $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is twice differentiable and strictly convex in s_i. $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is minimized by a strictly positive, unique, finite $s_i^*(\tau_i)$ that solves the first order condition, and is given by*

$$s_i^*(\tau_i) = \mathbb{E}[D_i(L, N_i, \tau_i)] + \sigma[D_i(L, N_i, \tau_i)]\Phi^{-1}\left(\frac{bT - \beta_i c(\tau_i)(1 + hT)}{bT}\right), \qquad (2.2)$$

where $\Phi^{-1}(\cdot)$ denotes the inverse of the standard normal distribution.

Proof. See Appendix. □

We insert the expression for the optimal turnaround stock levels from Eq. (2.2) into (CP) and (DP). However, this yields cost expressions that are not amenable for further analysis. To see this, let us consider $\tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$ for an arbitrary component $i \in I$. We note that $\tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$ is such that $c(\tau_i)$ is complexly embedded in the standard normal inverse function $\Phi^{-1}(\cdot)$. Therefore, we cannot determine the optimal reliability levels easily, as convexity or unimodality with respect to τ_i cannot be established. One solution may be to enumerate τ_i, but this may be expensive and time consuming to implement. Therefore, we propose another method that enables us to determine the optimal reliability levels approximately: we study asymptotically equivalent total LCC functions. We observe that the cost of downtime b is often high for the users of capital goods, typically in the order of tens of thousands U.S. dollars per hour (Parent, 2000; CNET News, 2001). In particular, if one considers time units of – say – months or even years, the downtime costs per time unit are enormous. Hence, we propose to study the asymptotic behavior of $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$ and $\tilde{\pi}(\tau_q, s_q^*(\tau_q), N_q, \beta_q)$ as b approaches infinity. Specifically, this means that we propose an approximate total LCC function for the common component model and for the dedicated components model. Both can be easily optimized and are asymptotically equivalent to the original total LCC functions as b tends to infinity. Moreover, the approximate functions enable us to analytically characterize and study our research questions, as we will see in the remainder of this chapter.

Before we present our approximate total LCC functions, we first add b as an argument to our cost functions, because we study the limit behavior with respect to b, i.e., we use $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)$ and $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b), N_q, \beta_q \mid b)$ because $s_i^*(\tau_i)$ also depends on b. If we now study $\tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)$, we observe that there exists a term $\Phi^{-1}\left(\frac{bT-\beta_i c(\tau_i)(1+hT)}{bT}\right)$ that is difficult to analyze with respect to the reliability τ_i. But if we substitute b by $b\beta_i c(\tau_i)$, then the term $\Phi^{-1}(\cdot)$ simplifies such that it is independent of τ_i. Consequently, the analysis of the LCC function of component $i \in I$ becomes tractable. Now, let the approximate total LCC functions for the dedicated and common components be $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))$ and $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b\beta_q c(\tau_q)), N_q, \beta_q \mid b\beta_q c(\tau_q))$, respectively. We prove that the two functions $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)$ and $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))$ are asymptotically equivalent as b tends to infinity. Furthermore, we also prove that the same asymptotic equivalence exists for $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b), N_q, \beta_q \mid b)$ and $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b\beta_q c(\tau_q)), N_q, \beta_q \mid b\beta_q c(\tau_q))$. Our techniques are similar to Huh et al. (2009) and Bijvank et al. (2014), and we use some of their results to show the asymptotic equivalence.

Lemma 2.2 *For each component $i \in I$, we have* $\lim_{b\to\infty} \frac{b\mathbb{E}[(D_i(L,N_i,\tau_i)-s_i^*(\tau_i,b))^+]}{s_i^*(\tau_i,b)} = 0$ *and* $\lim_{b\to\infty} \frac{b\beta_i c(\tau_i)\mathbb{E}[(D_i(L,N_i,\tau_i)-s_i^*(\tau_i,b\beta_i c(\tau_i)))^+]}{s_i^*(\tau_i,b\beta_i c(\tau_i))} = 0.$

Proof. See Appendix. □

We use Lemma 2.2 to establish our asymptotic equivalence result.

Theorem 2.1 *For given* $\tau_i \in (0, \overline{\tau})$*, the functions* $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i), N_i, \beta_i)$ *and* $\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))$ *are asymptotically equivalent as* $b \to \infty$*. That is,*

$$\lim_{b\to\infty} \frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)} = 1.$$

Furthermore, $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b\beta_q c(\tau_q)), N_q, \beta_q | b\beta_q c(\tau_q))$ *and* $\tilde{\pi}(\tau_q, s_q^*(\tau_q), N_q, \beta_q)$ *are asymptotically equivalent as* $b \to \infty$*. That is,*

$$\lim_{b\to\infty} \frac{\tilde{\pi}(\tau_q, s_q^*(\tau_q, b\beta_q c(\tau_q)), N_q, \beta_q \mid b\beta_q c(\tau_q))}{\tilde{\pi}(\tau_q, s_q^*(\tau_q, b), N_q, \beta_q \mid b)} = 1.$$

Proof. Fix $\tau_i \in (0, \overline{\tau})$ for all $i \in I$ and let $b \in (0, \infty)$ be sufficiently large to satisfy Assumption 2.1. By definition of $s_i^*(\tau_i, b)$, we find the following bounds by inserting suboptimal turnaround stock levels:

$$\begin{aligned}
&\frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b)} \\
&\qquad \le \frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)} \\
&\qquad \le \frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)}.
\end{aligned}$$

We define $k = \arg\max_{i\in J}\{s_i^*(\tau_i, b\beta_i c(\tau_i))\}$, and we rewrite the lower bound to

$$\begin{aligned}
&\frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b)} \\
&= \frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))/s_k^*(\tau_k, b\beta_k c(\tau_k))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b)/s_k^*(\tau_k, b\beta_k c(\tau_k))}.
\end{aligned}$$

Taking the limit as $b \to \infty$ implies that $s_i^*(\tau_i, b\beta_i c(\tau_i)) \to \infty$, and that any $\tau_i \in (0, \overline{\tau})$ satisfies Assumption 2.1. Consequently, we are not further concerned with satisfying Assumption 2.1. Each cost term in the numerator and denominator has a finite limit. This follows by using Lemma 2.2 and concluding that $\lim_{b\to\infty} \frac{b\beta_i c(\tau_i)\mathbb{E}[(D(L,N_i,\tau_i) - s_i^*(\tau_i, b\beta_i c(\tau_i)))^+]}{s_k^*(\tau_k, b\beta_k c(\tau_k))} = 0$, because $s_k^*(\tau_k, b\beta_k c(\tau_k)) \ge s_i^*(\tau_i, b\beta_i c(\tau_i))$ for all components $i \in J$. Hence, we obtain

$$\lim_{b\to\infty} \frac{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i\in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b)} = 1.$$

Now, we redefine $k = \arg\max_{i \in J}\{s_i^*(\tau_i, b)\}$ with a slight abuse of notation, and we rewrite the upper bound to

$$\frac{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)} = \frac{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b\beta_i c(\tau_i))/s_k^*(\tau_k, b)}{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)/s_k^*(\tau_k, b)}.$$

Again, taking the limit as $b \to \infty$ and applying Lemma 2.2, we obtain

$$\lim_{b \to \infty} \frac{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)} = 1.$$

By the sandwich theorem, we have $\lim_{b \to \infty} \frac{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))}{\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)} = 1.$

The proof of the asymptotic equivalence for the common component is analogous to the foregoing; we just have to omit the summations and replace all i and k by q. □

We now explore how well the approximate total LCC functions represent the actual total LCC functions. We introduce Testbed 2.1 to provide this comparison, and we use this Testbed 2.1 also in the remainder of this chapter.

Testbed 2.1 We consider a full factorial testbed, based on representative data for the semiconductor industry. We use a modified version of a well-established unit cost function for $c(\tau)$, see Mettas (2000) and Öner et al. (2010):

$$c(\tau) = p_1 + p_2 \exp\left(k \frac{\tau}{\overline{\tau} - \tau}\right), \quad p_1, p_2, k > 0, \quad 0 < \tau < \overline{\tau},$$

with $\overline{\tau} = 600$. Furthermore, we consider months as our time unit, i.e., τ_i, T, and L are in months; b is the cost per system down for one month; and h is a fraction per part per month. We generate a large testbed that considers 5,120 instances for dedicated components and 2,816 instances for the common component. We vary the following parameters on two levels, see Table 2.2.

h	r	b	T	L	p_1	p_2	k
0.015	0.2	100,000	180	3	500	100	1.0
0.030	0.3	1,000,000	360	4	5,000	1,000	2.0

Table 2.2: Parameter values for testbed

The values for b may seem excessive, but one should realize that these are the costs for one month downtime, and for capital goods these downtime costs are very large (roughly \$10,000 per hour). Thus, the value $b = \$100,000$ corresponds to a downtime costs of approximately \$140 per hour, which is very low for capital intensive systems. Furthermore, we consider two component families $|J| = 2$ and we vary the installed

base sizes and the relative unit cost factors. We set $\beta_1 = 1$ and vary β_2 as follows: $\beta_2 \in \{1; 1.05; 1.1; 1.15; 1.2; 1.25; 1.3\}$. For N_i, we let $\sum_{i \in J} N_i = N_1 + N_2 = 400$ for all instances, and vary one installed base size on three levels: $N_1 \in \{100, 200, 300\}$ and have $N_2 = 400 - N_1$. This yields $6 * 3 + 2 = 20$ possible instances (due to duplicates when $\beta_1 = \beta_2$) for each parameter combination from the parameters in Table 2.2. Hence, in total we have $20 \times 2^8 = 5,120$ instances.

For the common component we let $\beta_q \in \{1; 1.05; 1.1; \ldots; 1.5\}$ and $N_q = \sum_{i \in J} = 400$ is constant. This results in $11 \times 2^8 = 2,816$ instances for commonality. $\diamond$

We use Testbed 2.1 to see how well the approximate total LCC function of dedicated components compares to the original total LCC function of dedicated components. We also compare the approximate and original total LCC function of common components. For these comparisons, we enumerate the reliability levels over $\{1, 2, \ldots, 599\}$, and we let $\tilde{\tau}_i^*$ and τ_i^* correspond to the optimal reliability level for component $i \in I$ when using the original LCC function $\tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i \mid b)$ and the approximation $\tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i))$, respectively. Subsequently, we are interested in the relative cost differences for dedicated and common components

$$\Delta_J = \left(\frac{\sum_{i \in J} \tilde{\pi}(\tau_i^*, s_i^*(\tau_i^*, b\beta_i c(\tau_i^*)), N_i, \beta_i \mid b\beta_i c(\tau_i^*))}{\sum_{i \in J} \tilde{\pi}(\tilde{\tau}_i^*, s_i^*(\tilde{\tau}_i^*, b), N_i, \beta_i \mid b)} - 1 \right) \times 100\%, \quad \text{and}$$
$$\Delta_q = \left(\frac{\tilde{\pi}(\tau_q, s_q^*(\tau_q^*, b\beta_q c(\tau_q^*)), N_q, \beta_q \mid b\beta_q c(\tau_q^*))}{\tilde{\pi}(\tilde{\tau}_q^*, s_q^*(\tilde{\tau}_q^*, b), N_q, \beta_q \mid b)} - 1 \right) \times 100\%,$$

respectively. In particular, our interest goes out to the average, maximum, and minimum value of Δ_J and Δ_q. For Δ_J we find an average, maximum, and minimum value of 0.0028%, 0.0190%, and 0.0000%, respectively; and for Δ_q we find an average, maximum and minimum value of 0.0017%, 0.0095%, and 0.0000%. We observe that the average and maximum relative cost differences are very low. Therefore, we consider the approximate total LCC functions to be good approximations for the original total LCC functions.

Consequently, we propose to study $\sum_{i \in J} \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i | b\beta_i c(\tau_i))$ and $\tilde{\pi}(\tau_q, s_q^*(\tau_q, b\beta_q c(\tau_q)), N_q, \beta_q | b\beta_q c(\tau_q))$ in light of Theorem 2.1 and the results from Testbed 2.1. These approximate total LCC functions can be optimized easily and the satisfaction of Assumption 2.1 no longer depends on the reliability levels (once we consider the approximate total LCC functions). This results from the fact that the inverse of the standard normal cdf $\Phi^{-1}(\cdot)$ no longer depends on τ_i and thus is a constant. For brevity, we define

$$\pi(\tau_i, N_i, \beta_i) = \tilde{\pi}(\tau_i, s_i^*(\tau_i, b\beta_i c(\tau_i)), N_i, \beta_i \mid b\beta_i c(\tau_i)),$$

and we rewrite each $\pi(\tau_i, N_i, \beta_i)$ as follows, where $\phi(\cdot)$ and $\Phi^{-1}(\cdot)$ denote the standard normal pdf and the inverse of the standard normal cdf.

Lemma 2.3 *Inserting* $s_i^*(\tau_i, b\beta_i c(\tau_i))$ *into Eq.* (2.1) *yields*

$$\pi(\tau_i, N_i, \beta_i) = \beta_i c(\tau_i)\left(1 + \frac{rT + L(1+hT)}{\tau_i}\right) N_i + b\beta_i c(\tau_i) T \sqrt{\frac{\alpha N_i L}{\tau_i}} \phi\left(\Phi^{-1}\left(\frac{bT - 1 - hT}{bT}\right)\right). \tag{2.3}$$

Proof. See Appendix. □

We use Eq. (2.3) to derive the following cost minimization problems for the approximate formulations of the common component and of the dedicated components:

$$(CP') \quad \min_{\tau_q \in \mathbb{R}} \left\{\pi(\tau_q, N_q, \beta_q)\right\}, \text{ and}$$

$$(DP') \quad \min_{\boldsymbol{\tau} \in \mathbb{R}^{|J|}} \left\{\sum_{i \in J} \pi(\tau_i, N_i, \beta_i)\right\}.$$

The cost function of an arbitrary component $i \in I$ can be analyzed, because the dedicated components problem (DP') is separable in the components $i \in J$. Moreover, we make the following assumption with respect to $c(\tau)$.

Assumption 2.2 $c(\tau)$ satisfies the following: $\frac{c(\tau)}{\tau}$ is convex and $\frac{c(\tau)}{\sqrt{\tau}}$ is convex.

Assumption 2.2 is not very restrictive, because there exists a large class of functions that satisfy this assumption: polynomial functions with one constant term and all others terms being at least second order, and exponential forms, e.g. Mettas (2000) and Öner et al. (2010). Note that $c(\tau)$ from Testbed 2.1 satisfies Assumption 2.2.

Given Assumption 2.2 and after separation, we show that the cost function $\pi(\tau_i, N_i, \beta_i)$ for any component $i \in I$ is strictly convex if $c(\tau)$ satisfies Assumption 2.2. As a consequence, we can determine the optimal reliability levels τ_i^* for all components $i \in I$ easily by standard optimization techniques.

Lemma 2.4 *For each component* $i \in I$, $\pi(\tau_i, N_i, \beta_i)$ *is twice differentiable and strictly convex, and it is minimized by a positive, unique, finite* τ_i^*. *This* τ_i^* *solves the first order condition.*

Proof. See Appendix. □

Furthermore, we note that the optimal reliability levels τ_i^* are independent of the values of β_i. The cost function $\pi(\tau_i, N_i, \beta_i)$ is proportional to the unit cost $\beta_i c(\tau_i)$. As a consequence, there do not exist any economies of scale with respect to the unit production cost and neither with respect to β_i. Therefore, the relative cost factor β_i does not determine the optimal reliability level τ_i^*.

Secondly, each τ_i^* is determined by the size of the installed base N_i of component $i \in I$. This results from the fact that the production costs, the storage costs, and the repair costs are proportional to the installed base N_i. The costs for downtime and keeping safety stock are sub–linear in N_i due to service part pooling that exists when the installed base N_i increases or decreases.

2.4.2 Non–anticipating approach

For the models of the non–anticipating approach, $(\widehat{CP})$ and $(\widehat{DP})$, Assumption 2.1 is not relevant since we do not consider service parts. Let $\hat{\tau}_i^*$ minimize $(\widehat{DP})$ for component $i \in I$. Observing that $(\widehat{DP})$ is separable in the components $i \in J$ and given Assumption 2.2, we find all $\hat{\tau}_i^*$ efficiently by the result of Lemma 2.5.

Lemma 2.5 *The function $f(\tau) = c(\tau) + \frac{rc(\tau)T}{\tau}$ is twice differentiable and strictly convex, and it is minimized by a positive, unique, finite $\hat{\tau}^*$. This $\hat{\tau}^*$ solves the first order condition, and it satisfies $\hat{\tau}^* = \hat{\tau}_i^*$ for all $i \in I$.*

Proof. See Appendix. □

Finally, we remark that Lemma 2.5 implies that all $\hat{\tau}_i^*$ are independent of β_i and N_i as each term of $\hat{\pi}(\tau_i, N_i, \beta_i)$ is proportional to the unit production cost $\beta_i c(\tau_i)$ and also proportional to the installed base N_i. The independence of β_i follows by the same argument (absence of economies of scale with respect to β_i) that we presented in Section 2.4.1. The optimal reliability level $\hat{\tau}_i^*$ is independent on N_i, because the non–anticipating approach does not consider service parts in the decision. Hence, the costs for downtime and for keeping safety stock are not considered, and thus the LCC $\hat{\pi}(\tau_i, N_i, \beta_i)$ are proportional to N_i.

2.4.3 Comparing optimal reliability decisions

The optimal reliability levels under both approaches are different, and we study how different they are in this section. By using the results from the previous sections (asymptotic equivalence), we can easily determine the optimal reliability levels τ_i^* and $\hat{\tau}_i^*$ for all components $i \in I$ under the anticipating and non–anticipating approach, respectively. We use Testbed 2.1 to numerically study the differences between the optimal reliability levels. We are interested in the relative difference between the reliability levels τ_i^* and $\hat{\tau}_i^*$ for each component $i \in I$:

$$\Delta_i^r = \left(\frac{\tau_i^*}{\hat{\tau}_i^*} - 1 \right) \times 100\%.$$

The results show that the minimum and maximum relative reliability difference between both approaches (over all dedicated and common components) are 2.76% and 27.76%, respectively. The average reliability difference of the two dedicated components and the common components are 10.62%, 10.54%, and 9.57%, respectively. Hence, the OEM designs more reliable components when service part inventories are considered for the reliability decision. Furthermore, we observe that the difference in the optimal reliability levels can differ substantially between both approaches. These differences for the reliability decisions directly influence the intensity at which systems fail and thus may have a major effect on the performance and costs of after–sales services, e.g. service part provisioning. We do not elaborate on the numerical details, because our objective here is to illustrate that large differences exist in the optimal reliability levels between both approaches. Nevertheless, more detailed numerical results are presented and briefly discussed in Appendix 2.C.

2.5. Commonality decision

In this section, we study the optimal commonality decision for both approaches, under optimal reliability levels.

2.5.1 Non–anticipating approach

If the OEM uses the non–anticipating approach, we want to determine when he selects commonality, i.e., when does commonality yield lower LCC than dedicated components. Hence, our objective is to determine a condition such that $\hat{\pi}(\hat{\tau}_q^*, N_q, \beta_q) \leq \sum_{i \in J} \hat{\pi}(\hat{\tau}_i^*, N_i, \beta_i)$. We consider given values for the installed bases N_i and relative unit cost factors β_i for all components $i \in J$. Then, we derive a switching curve for β_q and we call this switching curve $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$.

Corollary 2.1 *Given $\boldsymbol{N}$ and $\boldsymbol{\beta}$, the non–anticipating approach selects commonality if and only if β_q is smaller than the weighted average of all β_i with $i \in J$. That is, the non–anticipating approach selects commonality if and only if $\beta_q \leq \hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta}) = \sum_{i \in J} \frac{\beta_i N_i}{N_q}$.*

Proof. See Appendix. □

Corollary 2.1 is a consequence of Lemma 2.5, and the former states that the non–anticipating approach selects commonality if and only if the relative unit cost factor of the common component β_q is less than the weighted average of the unit cost factors β_i of the dedicated components $i \in J$. In other words, commonality is attractive if and only if the common component's unit costs is less than the weighted average unit costs of the dedicated components. Hence, commonality is *not* always a good idea,

because the cost of the common component is a crucial determinant. Furthermore, Corollary 2.1 also implies that there cannot exist instances in which the common component's unit cost is higher than the unit costs of each dedicated component and commonality yields lower LCC than dedicated components. Thus, the OEM will not invest in a common component that is more expensive than any dedicated component if he uses the non–anticipating approach.

2.5.2 Anticipating approach

For the anticipating approach, we also determine a condition such that $\pi(\tau_q^*, N_q, \beta_q) \leq \sum_{i \in J} \pi(\tau_i^*, N_i, \beta_i)$. For each $i \in I$, the approximate LCC function $\pi(\tau_i, N_i, \beta_i)$ has the property that the optimal reliability level τ_i^* is independent of the relative unit cost factor β_i. Therefore, our objective is equivalent to determining a condition for β_q such that $\beta_q \pi(\tau_q^*, N_q, 1) \leq \sum_{i \in J} \pi(\tau_i^*, N_i, \beta_i)$, for given installed base sizes N_i and relative unit cost factors β_i for components $i \in J$ (note that $N_q = \sum_{i \in J} N_i$ is given). Hence, there exists a unique switching curve for β_q and we let $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ correspond to this switching curve. The anticipating approach selects commonality if and only if $\beta_q \leq \Theta(\boldsymbol{N}, \boldsymbol{\beta})$ with

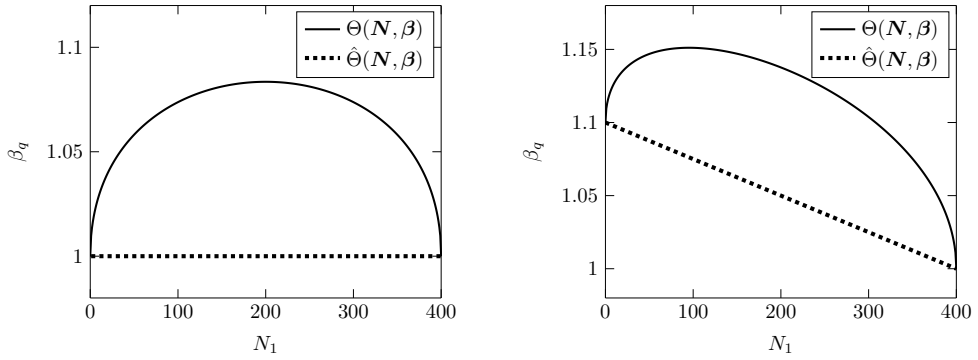
$$\Theta(\boldsymbol{N}, \boldsymbol{\beta}) = \frac{\sum_{i \in J} \pi(\tau_i^*, N_i, \beta_i)}{\pi(\tau_q^*, N_q, 1)}.$$

$\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ determines the commonality decision for the anticipating approach, and it can be interpreted as a measure of how expensive a common component can be and still yield lower total LCC than dedicated components. This implies that commonality does not necessarily reduce the costs: if the unit cost of a common component is considered, it may be more expensive to use commonality than using the alternative of dedicated components. We also remark that if we would study the original LCC function $\tilde{\pi}(\tau_i, s_i^*(\tau_i, b), N_i, \beta_i)$, there exist terms that are independent of β_i. This complicates the analytical characterization of a switching curve for commonality (if it even exists), because the optimal reliability levels depend on both $\boldsymbol{N}$ and $\boldsymbol{\beta}$. Thus, the asymptotic approximation from Section 2.4.1 enables us to analytically characterize a switching curve for commonality under the non–anticipating approach.

We shed a little more light on how $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ behaves with respect to the installed base sizes N_i and relative unit cost factors β_i. Consider an illustrative Example 2.1 with two dedicated components. We vary the installed base sizes $\boldsymbol{N}$ for two different vectors $\boldsymbol{\beta}$, and we obtain the results in Figure 2.3. Furthermore, we also illustrate $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ in Figure 2.3.

Example 2.1 For this numerical example, we consider a representative parameter setting for an OEM in the semiconductor industry. Let $|J| = 2$, $h = 0.03$ per part per month, $r = 0.2$ per repair, $L = 3$ months, $T = 360$ months, $b = \$1 \times 10^6$ per month per system down, and $\alpha = 1$. Moreover, we consider the installed base $N_1 \in \{1, \ldots, 399\}$

and let $N_1 + N_2 = 400$. Furthermore, we use $c(\tau) = 5,000 + 1,000 \exp\left(\frac{\tau}{600-\tau}\right)$ in \$ per unit, where $\tau \in (0, 600)$. ◇



(a) $\beta_1 = \beta_2 = 1$ *(b)* $\beta_1 = 1, \beta_2 = 1.1$

Figure 2.3: Illustration of $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ and $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ from Example 2.1

The results in Figure 2.3 for $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ are straightforward as the switching curve is the weighted average of β_1 and β_2. Hence, we focus our attention on $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ in the remainder.

Using a common component has the advantage of exploiting service parts pooling, i.e., offer the same performance with fewer service parts on stock. As a result, the costs can be reduced when using common components compared to using dedicated components. However, we typically observe that common components are more expensive than each of the dedicated components (van Mieghem, 2004), i.e., $\beta_q \geq \beta_i$ for all components $i \in J$. Figure 2.3 illustrates the trade off between the cost benefit of pooling service parts and the increased unit cost of a common component β_q. We see that – in most cases – the unit cost of a common component in terms of β_q can be substantially higher (compared to β_i) and still result in lower life cycle costs due to the cost reduction that results from service parts pooling.

Furthermore, if we consider the results of the symmetrical dedicated components in Figure 2.3a, the cost advantage resulting from service parts pooling increases as the difference between the installed base sizes becomes smaller. This means that, when N_1 and N_2 are closer together, the unit cost of a common component in terms of β_q can increase more and still result lower LCC than dedicated components. The symmetry of $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ follows from the symmetry that exists for the dedicated components ($\beta_1 = \beta_2$). Hence, the maximum value for $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ is attained when both installed bases are identical, because this results in the largest pooling effect for service parts, and thus allows the unit cost of a common component to become most expensive. Hence, we say that commonality is most attractive when the installed base sizes are equal.

If the dedicated components are not symmetrical ($\beta_1 = 1$ and $\beta_2 = 1.1$), we obtain Figure 2.3b. Using a common component still profits from the cost benefit that results from pooling the service parts. Consequently, the common component can increase the unit cost (in terms of β_q) and still result in lower LCC. However, $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ is no longer symmetrical but skewed, because the pooled dedicated components have different values for β_i. Furthermore, $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ is skewed such that commonality is most attractive – $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ is maximal – when the dedicated component i with the largest β_i also has the largest installed base N_i. If the installed base of expensive parts N_2 is relatively large (and N_1 is small), the total costs of dedicated components is also relatively large. Consequently, the unit cost of a common component (in terms of β_q) can increase more and still result in lower life cycle costs, in part also due to the cost benefit of service part pooling. Hence, commonality is most attractive when a more expensive component has a larger installed base than a cheaper component.

Proving our observations for when commonality is most attractive (maximum value of $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$) is difficult, because a change in the installed base size N_i induces a change in the optimal reliability level τ_i^*. Therefore, we propose to study an upper bound $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ for $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$. This upper bound is obtained by substituting the optimal reliability levels τ_i^* by the suboptimal reliability level τ_q^* in the numerator of $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$. This yields the following switching curve upper bound for commonality:

$$\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta}) = \frac{\sum_{i \in J} \pi(\tau_q^*, N_i, \beta_i)}{\pi(\tau_q^*, N_q, 1)}.$$

Using Testbed 2.1, we study how tight this upper bound $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ is. We consider the relative difference $\left(\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})/\Theta(\boldsymbol{N}, \boldsymbol{\beta}) - 1\right) \times 100\%$, and find that the average, maximum, and minimum difference between $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ and $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ are 0.0049%, 0.0207%, and 0.0000%, respectively. The differences are very small, which results from the fact that the optimal reliability levels τ_i^* are close to τ_q^*. Therefore, it is likely that the unit production cost in terms of β_i and the installed base N_i mainly determine how attractive commonality is.

As the differences between $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ and $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ are very small, we study the upper bound $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ in more detail in the remainder of this section. We prove – for $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ – that commonality is most attractive when the installed base sizes are identical if $\beta_i = \beta_j$ for all components $i \in J$. Secondly, we also prove that commonality is most attractive when N_i follows the same ordering as β_i, if the β_i are not identical.

Proposition 2.1

(a) If $\beta_i = \beta_j$ for all components $i, j \in J$, $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ increases when the difference in installed base sizes decreases. That is, when $\beta_i = \beta_j$, $\forall i, j \in J$ and $\sum_{i \in J} N_i = N_q$ for some $N_q \in \mathbb{N}$, if there exist $j, k \in J$ such that $N_j - N_k > 1$ then $\tilde{\Theta}(\boldsymbol{N} - e_j + e_k, \boldsymbol{\beta}) \geq \tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$, where e_j denotes the indicator vector of component $j \in J$.

(b) For any β_i and relaxed integrality of N_i for all components $i \in J$, $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ increases when the installed base sizes are ordered the same way as the relative unit cost factors. That is, if $l = |J|$ and $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_l$, then the vector $\boldsymbol{N^*} \in \mathbb{R}^l$ that maximizes $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ such that $\sum_{i \in J} N_i^* = N_q$ for some $N_q \in \mathbb{N}$ satisfies $N_1^* \leq N_2^* \leq \ldots \leq N_l^*$.

Proof. See Appendix. □

The result of Proposition 2.1 proves our observations for the upper bound $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$. These results can likely be extrapolated to the actual threshold $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$, because the upper bound is tight with a worst case gap of 0.0207%. This indicates that commonality becomes most attractive – under the anticipating approach – for equally sized installed bases if the relative unit costs are equal; if the relative unit costs differ, commonality is most attractive when components with higher unit costs also have a larger installed base.

2.5.3 Comparing commonality decisions

Next, we compare the difference in the commonality decision between both approaches. We compare the switching curve of the anticipating approach $\Theta(\boldsymbol{N}, \boldsymbol{\beta})$ to the switching curve of the non–anticipating approach $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$.

Theorem 2.2 *The anticipating approach selects commonality strictly more than the non–anticipating approach, i.e., $\Theta(\boldsymbol{N}, \boldsymbol{\beta}) > \hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$.*

Proof. For our claim, it suffices to prove that $\Theta(\boldsymbol{N}, \boldsymbol{\beta}) > \sum_{i \in J} \frac{\beta_i N_i}{N_q}$ as $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta}) = \sum_{i \in J} \frac{\beta_i N_i}{N_q}$. Let $\beta_q = \sum_{i \in J} \frac{N_i \beta_i}{N_q}$ and consider $\pi(\tau_q^*, N_q, \beta_q)$ to obtain

$$\begin{aligned}
\pi(\tau_q^*, N_q, \beta_q) &= \sum_{i \in J} \frac{N_i}{N_q} \beta_i c(\tau_q^*) \left(1 + \frac{rT + L(1+hT)}{\tau_q^*}\right) N_q \\
&\quad + \sum_{i \in J} \frac{N_i}{N_q} b \beta_i c(\tau_q^*) T \sqrt{\frac{\alpha N_q L}{\tau_q^*}} \phi\left(\Phi^{-1}\left(\frac{bT - 1 - hT}{bT}\right)\right) \\
&\leq \sum_{i \in J} \frac{N_i}{N_q} \beta_i c(\tau_i^*) \left(1 + \frac{rT + L(1+hT)}{\tau_i^*}\right) N_q \\
&\quad + \sum_{i \in J} \frac{N_i}{N_q} b \beta_i c(\tau_i^*) T \sqrt{\frac{\alpha N_q L}{\tau_i^*}} \phi\left(\Phi^{-1}\left(\frac{bT - 1 - hT}{bT}\right)\right) \\
&= \sum_{i \in J} \beta_i c(\tau_i^*) \left(1 + \frac{rT + L(1+hT)}{\tau_i^*}\right) N_i
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{i\in J}\frac{N_i}{N_q}b\beta_i c(\tau_i^*)T\sqrt{\frac{\alpha N_q L}{\tau_i^*}}\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&<\sum_{i\in J}\beta_i c(\tau_i^*)\left(1+\frac{rT+L(1+hT)}{\tau_i^*}\right)N_i\\
&+\sum_{i\in J}b\beta_i c(\tau_i^*)T\sqrt{\frac{\alpha N_i L}{\tau_i^*}}\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&=\sum_{i\in J}\pi(\tau_i^*,N_i,\beta_i),
\end{aligned}
$$

where the first inequality follows from inserting suboptimal values τ_i^* instead of the optimal τ_q^*. The last inequality follows from the fact that if $N_i > 0$, then $\frac{N_i}{N_q}\sqrt{N_q} = \sqrt{N_i}\sqrt{\frac{N_i}{N_q}} < \sqrt{N_i}$ for all $i \in J$. As $\beta_q = \sum_{i\in J}\frac{N_i\beta_i}{N_q}$ is the maximum value for β_q such that commonality is selected under the non–anticipating approach and β_q is such that $\pi(\tau_q^*, N_q, \beta_q) < \sum_{i\in J}\pi(\tau_i, N_i, \beta_i)$, the anticipating approach still selects commonality. Hence, we have that $\Theta(\boldsymbol{N},\boldsymbol{\beta}) > \sum_{i\in J}\frac{N_i\beta_i}{N_q} = \hat{\Theta}(\boldsymbol{N},\boldsymbol{\beta})$, where the final equality holds by Corollary 2.1. □

Theorem 2.2 proves that the anticipating approach favors commonality strictly more than the non–anticipating approach. The former approach incorporates the service parts pooling effects in its decision making, while the latter approach does not. This causes the non–anticipating approach to underestimate the attractiveness of commonality and it may lead to suboptimal commonality decisions that in turn may result in higher total LCC. Furthermore, the result from Theorem 2.2 confirms previous findings on the beneficial inventory effects of commonality; see for example Baker et al. (1986), Hillier (2000), van Mieghem (2004), and Song and Zhao (2009). Our result proves that these insights also hold in settings where LCC and reliability optimization are considered.

Besides the analytical difference between the switching curves of both approaches, we also numerically explore the difference in the commonality decision between the two approaches. We are primarily interested in the cost increase of the unit cost for a common component such that both approaches still result in a different commonality decision. This is equivalent to considering the relative increase in the switching curves between the two approaches, given by $\left(\Theta(\boldsymbol{N},\boldsymbol{\beta})/\hat{\Theta}(\boldsymbol{N},\boldsymbol{\beta}) - 1\right) \times 100\%$. We use Testbed 2.1 to find that the average, maximum, and minimum values are 5.41%, 9.59%, and 2.33%, respectively. Thus, we see that the two approaches can still yield a different commonality decision, even when the common component's unit cost increases substantially by as much as 9.59%. This difference underlines the importance of considering service part inventories in the commonality decision, and it implies that ignoring them in such a decision underestimates the attractiveness of commonality. More extensive numerical results and discussions are presented in

Appendix 2.C.

2.6. Cost effects

In this final analysis part of the chapter, we study how much of the LCC we can save when we consider service parts in the design decision in terms of commonality and reliability (RQ3). Our objective is to compare the costs of the anticipating and the non–anticipating approach. Since the anticipating approach includes more LCC aspects (service parts), we evaluate the decision of the non–anticipating approach based on the total LCC functions $\pi(\tau_q, N_q, \beta_q)$ and $\sum_{i\in J}\pi(\tau_i, N_i, \beta_i)$ of the anticipating approach. We are interested in the relative cost difference between both approaches, defined by

$$\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q) = \left(\frac{\gamma\pi(\hat{\tau}_q^*, N_q, \beta_q) + (1-\gamma)\sum_{i\in J}\pi(\hat{\tau}_i^*, N_i, \beta)}{\min\{\pi(\tau_q^*, N_q, \beta_q), \sum_{i\in J}\pi(\tau_i^*, N_i, \beta_i)\}} - 1\right) \times 100\%,$$

where γ is a binary variable such that $\gamma = 1$ if $\hat{\pi}(\hat{\tau}_q^*, N_q, \beta_q) \leq \sum_{i\in J}\hat{\pi}(\hat{\tau}_i^*, N_i, \beta_i)$ and $\gamma = 0$ otherwise. Note that we use γ to indicate whether the non–anticipating approach selects commonality, and we evaluate this decision under the LCC expressions of the anticipating approach $\pi(\hat{\tau}_q^*, N_q, \beta_q)$ and $\sum_{i\in J}\pi(\hat{\tau}_i^*, N_i, \beta)$. Then, we have that $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q) \geq 0$ for all $\boldsymbol{N}$, $\boldsymbol{\beta}$ and β_q, because we compare the costs of the optimal decisions under the anticipating approach to the (suboptimal) decisions made under the non–anticipating approach. Consequently, the anticipating approach results in lower LCC than the non–anticipating approach.

The reason for this LCC reduction stems from the two advantages that the anticipating approach has over the non–anticipating approach: the first benefit is that (i) the anticipating approach considers service parts in the commonality decision. Therefore, it takes the advantageous effect of service parts pooling into account. This again confirms results from previous research that common components are beneficial for service parts pooling (Baker et al., 1986; Hillier, 2000; Song and Zhao, 2009). Our findings do not only confirm previous results, but also show that the anticipating approach (considering service parts) reduces the LCC for *any* installed base size vector $\boldsymbol{N}$ and for *any* unit costs β_i of component $i \in I$, i.e., for any $\boldsymbol{N}$, $\boldsymbol{\beta}$, and β_q, we have $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q) \geq 0$. That is, for any instance – even when the common component is very expensive – the anticipating approach reduces the LCC compared to the non–anticipating approach. Secondly, (ii) as the anticipating approach considers service parts, the OEM makes a conscious decision on the substitution effect between reliability and service parts (Öner et al., 2010; Kim et al., 2017). This enables the OEM to further reduce the LCC by taking the anticipating approach.

Despite the result of Theorem 2.2, we have no indication how large the LCC reduction is. Therefore, we numerically study $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ based on Testbed 2.1.

Testbed 2.1 (continued) For each instance of dedicated components, we consider $\beta_q \in \{1; 1.05; 1.1; \ldots; 1.5\}$ in order to determine $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$. This increases the number of instances from 5,120 to 56,320. ◇

For Testbed 2.1, we obtain the average, maximum, and minimum value of the cost difference $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ of 0.80%, 10.67%, and 0.07%, respectively. We see that the cost differences between both approaches can be substantial, although the average value is relatively low. Thus, there exist instances in which the LCC can become 10% more expensive if the OEM omits the service parts consideration from the design decision in terms of commonality and reliability. Such a 10% difference is an increase of the *total* life cycle costs and may have a direct detrimental effect on the OEM's profitability. Hence, the OEM is urged to consider service parts in the commonality and reliability decisions.

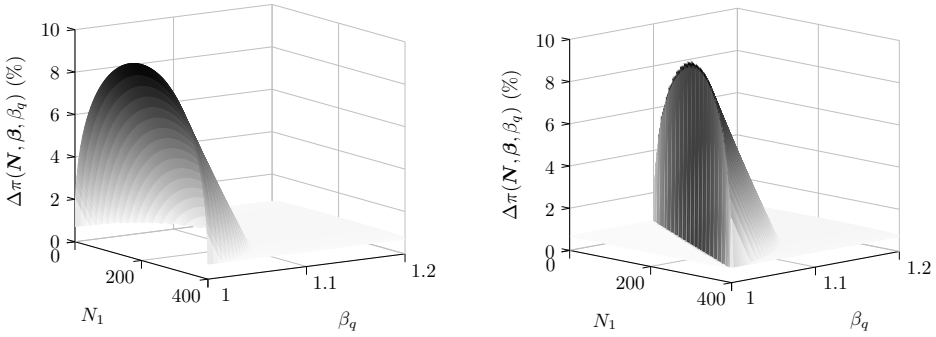
Yet, we remark that the average value of the relative cost difference is fairly low ($< 1\%$). This follows from the fact that our testbed contains 50,704 instances (out of the 56,320 instances) in which the anticipating and non–anticipating result in the same commonality decision. In the 5,616 instances wherein the anticipating approach selects commonality and the non–anticipating approach does not, we observe an average, maximum, and minimum LCC difference of 3.12%, 10.46%, and 0.09%. Therefore, we see that the LCC difference significantly increases when the commonality decision is different for both approaches, thereby further illustrating the need of considering service parts for the commonality and reliability decisions.

In the other instances, the non–anticipating and anticipating approach yield the same commonality decision. For these instances, we find an average, maximum, and minimum LCC difference of 0.54%, 10.67%, and 0.07%, respectively. Although the average LCC difference is low, there still exist instances in which this difference can be as large as 10%. Such a large difference is a result of the reliability decisions, since the commonality decision is the same for both approaches. To avoid large LCC differences that may be detrimental to the OEM's profitability, it is crucial to consider service parts in the commonality and reliability decisions. A more extensive numerical study considering the LCC difference is presented in Appendix 2.C.

In addition to results of Testbed 2.1, we also study Example 2.1 in more detail. This enables us to graphically illustrate cases in which the relative cost difference $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ is large.

Example 2.1 (continued) We consider $\beta_q \in \{1; 1.001; 1.002; \ldots; 1.2\}$. ◇

We use Example 2.1 to generate the numerical results from Figure 2.4. The elevated 'surfaces' in Figure 2.4 show the larger cost differences. If we now look at Figure 2.4 from the top, we obtain Figure 2.3. This observation implies that the larger cost differences are typically observed when the commonality decision differs between the non–anticipating and the anticipating approach. Moreover, we see that the



(a) $\beta_1 = \beta_2 = 1$ *(b)* $\beta_1 = 1, \beta_2 = 1.1$

Figure 2.4: Numerical illustration of $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ for Example 2.1

largest cost differences occur when β_q is just larger than the switching curve of the non–anticipating approach. So, if the non–anticipating approach selects dedicated components only because it is slightly cheaper than commonality, the actual total LCC may become substantially higher, even up to 10% (as we saw in Testbed 2.1). Therefore, we strongly urge OEMs to consider service parts in their design decisions, as this enables them to make better decisions that result in lower LCC, at the expense of very little implementation complexity.

2.7. Conclusion

We considered an OEM that is responsible for a number of systems because he has closed service contracts with his customers. Hence, the OEM is reponsible for a major part of a system life cycle and thus interested in minimizing the LCC. We studied the effect of service parts on the design decisions that are made by an OEM, and its consequences on the LCC. The OEM determines whether to make components common and what reliability the components have. We considered an approach that neglects service parts in the design decision and called this the non–anticipating approach. This approach is mostly used in practice, because a design department carries no responsibility for after–sales performance such as service parts planning. On the other hand, we also studied an approach that does consider service parts in the design decision, and called this the anticipating approach.

For each approach, we formulated two LCC models: one commonality model, and one model for the alternative of dedicated components. The optimization of the models for the anticipating approach was intractable, and therefore we proposed two approximate models that are asymptotically equivalent as the cost of system downtime tends to infinity. The benefit of studying these approximate models (for the anticipating

approach) is that it enables us to efficiently determine the optimal reliability levels, and we are able to obtain analytical results for the commonality decision. The optimization for the non–anticipating approach was rather straightforward.

We studied the optimal reliability decision for the non–anticipating and anticipating approach. We proved that the optimal reliability levels can be determined by straightforward optimization. Moreover, we numerically illustrated that considering service parts in the reliability decision is essential and it increases the optimal reliability levels by as much as 27%, and on average by roughly 10%. Such differences are large and can have major effects on the after–sales operations and costs. Therefore, engineers should be incentivized to consider service parts for the reliability decision, if the OEM's objective is to minimize the life cycle costs. Secondly, we focused on the commonality decision for both approaches. We analytically characterized – for each approach – a switching curve based on the unit cost of a common component, and this curve determines the commonality decision. This curve indicates that commonality does not always reduce the costs, because it depends on the cost of a common component: if a common component is relatively expensive (compared to dedicated components), commonality is not attractive. We also found that the anticipating approach selects commonality strictly more than the non–anticipating approach. In addition to the analytical result, we numerically saw that the decision between both approaches can differ, even if the unit cost of a common component increases by as much as 9.59%. So, service parts should be considered if a good commonality decision has to be made: the OEM can use a significantly more expensive common component and still obtain lower life cycle costs. We also studied the life cycle costs effect of considering service parts in design decisions. We proved that the non–anticipating approach yields higher LCC than the anticipating approach. This occurs because the non–anticipating approach does not exploit the service parts pooling effect and the substitution effect between service parts and reliability. Our numerical analysis indicated that if we pursue the non–anticipating approach, we may end in scenarios in which we incur 10% higher life cycle costs, which may be detrimental to an OEM's profitability. Hence, if the OEM wants to improve its profitability, it should encourage his design engineers to consider service parts in the design decisions.

2.A. Proofs

2.A.1 Proof of Lemma 2.1

Let us first write:

$$\begin{aligned}\tilde{\pi}(\tau_i, s_i, N_i, \beta_i) &= \beta_i c(\tau_i)(N_i + s_i) + hs_i T\beta_i c(\tau_i) + r\beta_i c(\tau_i)\frac{N_i T}{\tau_i} \\ &\quad + bT\int_{s_i}^{\infty}(x - s_i)f_i(x)dx \\ &= \beta_i c(\tau_i)(N_i + s_i) + hs_i T\beta_i c(\tau_i) + r\beta_i c(\tau_i)\frac{N_i T}{\tau_i} \\ &\quad + bT\mathbb{E}[D_i(L, N_i, \tau_i)] - bTs_i + bT\int_0^{s_i}(s_i - x)f_i(x)dx,\end{aligned} \tag{2.4}$$

with $f_i(x)$ the pdf of $D_i(L, N_i, \tau_i)$. From Leibniz' rule, we obtain $\frac{\partial\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i} = \beta_i c(\tau_i) + \beta_i c(\tau_i)hT - bT + bT\int_0^{s_i} f_i(x)dx = \beta_i c(\tau_i) + \beta_i c(\tau_i)hT - bT + bTF(s_i)$, with $F(s_i)$ the cdf of $D_i(L, N_i, \tau_i)$. Applying Leibniz' rule again, we obtain the second order derivative $\frac{\partial^2\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial {s_i}^2} = bTf_i(s_i) > 0$, as $b, T > 0$, and $f_i(s_i) > 0$ by definition of the pdf. Hence, $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is twice differentiable and strictly convex in s_i. Next, we prove the existence of a positive, unique, finite $s_i^*(\tau_i)$ that solves the first order condition.

(i) First, we prove that $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is strictly decreasing at $s_i = 0$. Consider the derivative $\frac{\partial\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i}$ as $s_i = 0$:

$$\begin{aligned}\left.\frac{\partial\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i}\right|_{s_i=0} &= \beta_i c(\tau_i) + \beta_i c(\tau_i)hT - bT + bTF_i(s_i) = \beta_i c(\tau_i) + \beta_i c(\tau_i)hT \\ &\quad - bT + bT\Phi\left(-\frac{\mathbb{E}[D_i(L, N_i, \tau_i)]}{\sigma[D_i(L, N_i, \tau_i)]}\right) \\ &\leq \beta_i c(\tau_i) + \beta_i c(\tau_i)hT - \frac{bT}{2} \\ &< 0,\end{aligned}$$

where the second equality follows because we consider normally distributed demand during L, and the first inequality follows because $\Phi\left(-\frac{\mathbb{E}[D_i(L, N_i, \tau_i)]}{\sigma[D_i(L, N_i, \tau_i)]}\right) \leq 1/2$, with $\Phi(\cdot)$ denoting the standard normal cdf. The final inequality follows from Assumption 2.1. Hence, we conclude that $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is strictly decreasing at $s_i = 0$.

(ii) Let us now prove that $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is strictly increasing as s_i tends to infinity. Consider the derivative $\frac{\partial\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i}$ as $s_i \to \infty$, i.e., $\lim_{s_i\to\infty}\frac{\partial\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i} = \lim_{s_i\to\infty}\{\beta_i c(\tau_i) + \beta_i c(\tau_i)hT - bT + bTF_i(s_i)\} = \beta_i c(\tau_i)(1 + hT) > 0$, where the last equality follows from the definition of $F_i(s_i)$, i.e., $\lim_{s_i\to\infty} F_i(s_i) = 1$. The inequality

follows from $c(\tau_i) > 0$ for all $\tau_i \in (0, \overline{\tau})$, and $\beta_i, h, T > 0$. Thus, $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$ is strictly increasing as s_i tends to infinity.

By combining *(i)*, *(ii)* and the strict convexity of $\tilde{\pi}(\tau_i, s_i, N_i, \beta_i)$, there exists a positive, unique, finite optimum $s_i^*(\tau_i)$ that solves the first order condition. Exploiting the properties of the normal distribution, and standardization yields

$$s_i^*(\tau_i) = \mathbb{E}[D_i(L, N_i, \tau_i)] + \sigma[D_i(L, N_i, \tau_i)]\Phi^{-1}\left(\frac{bT - \beta_i c(\tau_i)(1 + hT)}{bT}\right).$$

2.A.2 Proof of Lemma 2.2

For given $\tau_i \in (0, \overline{\tau})$, we let b be such that we satisfy Assumption 2.1. For each $i \in I$ we have

$$\begin{aligned}
0 &\leq \frac{b\mathbb{E}[(D_i(L, N_i, \tau_i) - s_i^*(\tau_i, b))^+]}{s_i^*(\tau_i, b)} \\
&= \frac{b\mathbb{P}\left[D_i(L, N_i, \tau_i) > s_i^*(\tau_i, b)\right]\mathbb{E}\left[D_i(L, N_i, \tau_i) - s_i^*(\tau_i, b) \mid D_i(L, N_i, \tau_i) > s_i^*(\tau_i, b)\right]}{s_i^*(\tau_i, b)} \\
&= \frac{\beta_i c(\tau_i)(1 + hT)}{T} \times \frac{\mathbb{E}\left[D_i(L, N_i, \tau_i) - s_i^*(\tau_i, b) \mid D_i(L, N_i, \tau_i) > s_i^*(\tau_i, b)\right]}{s_i^*(\tau_i, b)},
\end{aligned}$$

where the first inequality follows from Eq. (2.4) in the proof of Lemma 2.1. The second equality follows from the definition of the optimal turnaround stock level. That is, $s_i^*(\tau_i, b)$ satisfies $\frac{\partial \tilde{\pi}(\tau_i, s_i, N_i, \beta_i)}{\partial s_i} = 0$, which implies $\mathbb{P}[D_i(L, N_i, \tau_i) > s_i^*(\tau_i, b)] = \frac{\beta_i c(\tau_i)(1+hT)}{bT}$ by the right continuity of the distribution function (Huh et al., 2009, p. 409). Then, for the limit of $b \to \infty$, Assumption 2.1 is satisfied for any finite $\tau_i \in (0, \overline{\tau})$ and thus we obtain

$$\begin{aligned}
0 &\leq \lim_{b\to\infty} \frac{b\mathbb{E}[(D_i(L, N_i, \tau_i) - s_i^*(\tau_i, b))^+]}{s_i^*(\tau_i, b)} \\
&\leq \lim_{b\to\infty} \frac{\beta_i c(\tau_i)(1 + hT)}{T} \times \frac{\mathbb{E}\left[D_i(L, N_i, \tau_i) - s_i^*(\tau_i, b) \mid D_i(L, N, \tau_i) > s_i^*(\tau_i, b)\right]}{s_i^*(\tau_i, b)} = 0.
\end{aligned}$$

The equality follows from the fact that $s_i^*(\tau_i, b) \to \infty$ as $b \to \infty$ and because $\frac{\mathbb{E}[D_i(L,N_i,\tau_i) - s_i^*(\tau_i,b) \mid D_i(L,N_i,\tau_i) > s_i^*(\tau_i,b)]}{s_i^*(\tau_i,b)} \to 0$ as $b \to \infty$ due to the increasing failure rate of a normal distribution, see Huh et al. (2009, p. 409).

2.A.3 Proof of Lemma 2.3

Let us write $\pi(\tau_i, N_i, \beta_i)$ in terms of the normalized loss function of the normal distribution, with $\hat{s}_i = s_i^*(\tau_i, b\beta_i c(\tau_i))$.

$$\pi(\tau_i, N_i, \beta_i) = \beta_i c(\tau_i)(N_i + \hat{s}_i) + h\hat{s}_i T\beta_i c(\tau_i) + r\beta_i c(\tau_i)\frac{N_i T}{\tau}$$

$$+b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\left\{\phi\left(\frac{\hat{s}_i-\mathbb{E}[D_i(L,N_i,\tau_i)]}{\sigma[D_i(L,N_i,\tau_i)]}\right)\right.$$
$$\left.-\frac{\hat{s}_i-\mathbb{E}[D_i(L,N_i,\tau_i)]}{\sigma[D_i(L,N_i,\tau_i)]}\left(1-\Phi\left(\frac{\hat{s}_i-\mathbb{E}[D_i(L,N_i,\tau_i)]}{\sigma[D_i(L,N_i,\tau_i)]}\right)\right)\right\},$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal pdf and cdf, respectively. Next, we substitute $\hat{s}_i$ by $s_i^*(\tau_i, b\beta_i c(\tau_i))$ and simplify:

$$\begin{aligned}
&\pi(\tau_i,N_i,\beta_i))\\
&=\beta_i c(\tau_i)N_i+\beta_i c(\tau)(1+hT)\left(\mathbb{E}[D_i(L,N_i,\tau_i)]\right.\\
&\quad\left.+\sigma[D_i(L,N_i,\tau_i)]\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)+r\beta_i c(\tau_i)\frac{N_iT}{\tau}\\
&\quad+b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&\quad-b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\left(1-\Phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\right)\\
&=\beta_i c(\tau_i)N_i+\beta_i c(\tau_i)(1+hT)\left(\mathbb{E}[D_i(L,N_i,\tau_i)]\right.\\
&\quad\left.+\sigma[D_i(L,N_i,\tau_i)]\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)+r\beta_i c(\tau_i)\frac{N_iT}{\tau_i}\\
&\quad+b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&\quad-\beta_i c(\tau_i)(1+hT)\sigma[D_i(L,N_i,\tau_i)]\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\\
&=\beta_i c(\tau_i)N_i+\beta_i c(\tau_i)(1+hT)\mathbb{E}[D_i(L,N_i,\tau_i)]+r\beta_i c(\tau_i)\frac{N_iT}{\tau_i}\\
&\quad+b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&=\beta_i c(\tau_i)\left(1+\frac{rT+L(1+hT)}{\tau_i}\right)N_i\\
&\quad+b\beta_i c(\tau_i)T\sigma[D_i(L,N_i,\tau_i)]\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right)\\
&=\beta_i c(\tau_i)\left(1+\frac{rT+L(1+hT)}{\tau_i}\right)N_i+b\beta_i c(\tau_i)T\sqrt{\frac{\alpha N_iL}{\tau_i}}\phi\left(\Phi^{-1}\left(\frac{bT-1-hT}{bT}\right)\right).
\end{aligned}$$

2.A.4 Proof of Lemma 2.4

All terms are twice differentiable by assumption, and thus $\pi(\tau_i, N_i, \beta_i)$ is twice differentiable. We have

$$\frac{d^2\pi(\tau_i,N_i,\beta_i)}{d\tau_i^2}=c''(\tau_i)N_i\beta_i+\beta_i(rT+L(1+hT))N_i\left(\frac{c''(\tau_i)}{\tau_i}-2\frac{c'(\tau_i)}{\tau_i^2}+2\frac{c(\tau_i)}{\tau_i^3}\right)$$

$$
\begin{aligned}
&+\left(c''(\tau_i)\tau_i^{-1/2}-c'(\tau_i)\tau_i^{-3/2}+\frac{3}{4}c(\tau_i)\tau_i^{-5/2}\right)\\
&\times\beta_i bT\phi\left(\Phi^{-1}\left(\frac{bT-1-hT)}{bT}\right)\right)\sqrt{\alpha N_i L}\\
&=c''(\tau_i)N_i\beta_i+\beta_i(rT+L(1+hT))\frac{N_i}{\tau_i^3}\left(\tau_i^2c''(\tau_i)-2\tau_i c'(\tau_i)+2c(\tau_i)\right)\\
&+\tau_i^{-5/2}\left(\tau_i^2c''(\tau_i)-\tau_i c'(\tau_i)+\frac{3}{4}c(\tau_i)\right)\\
&\times\beta_i bT\phi\left(\Phi^{-1}\left(\frac{bT-1-hT)}{bT}\right)\right)\sqrt{\alpha N_i L}\\
&>0,
\end{aligned}
$$

since $c''(\tau_i)>0$ (by assumption) and $c(\tau_i)$ satisfying Assumption 2.2. The latter can be seen by the second order derivative test applied to the conditions in Assumption 2.2. Hence, $\pi(\tau_i,N_i,\beta_i)$ is twice differentiable and strictly convex in τ_i. The next step is to show that there exists a positive, unique, finite τ_i^* that minimizes $\pi(\tau_i,N_i,\beta_i)$, and that this τ_i^* solves the first order condition.

(i) $\pi(\tau_i,N_i,\beta_i)$ is strictly decreasing for $\tau_i\downarrow 0$. The derivative of $\pi(\tau_i,N_i,\beta_i)$ is given by

$$
\begin{aligned}
&\frac{d\pi(\tau_i,N_i,\beta_i)}{d\tau_i}\\
&=c'(\tau_i)N_i\beta_i+(rT+L(1+hT))N_i\beta_i\left(\frac{c'(\tau_i)}{\tau_i}-\frac{c(\tau_i)}{\tau_i^2}\right)\\
&\quad+\left(c'(\tau_i)\tau_i^{-1/2}-\frac{1}{2}c(\tau_i)\tau_i^{-3/2}\right)\beta_i bT\phi\left(\Phi^{-1}\left(\frac{bT-1-hT)}{bT}\right)\right)\sqrt{\alpha N_i L}.
\end{aligned}
$$

Rewriting $\frac{c'(\tau_i)}{\tau_i}-\frac{c(\tau_i)}{\tau_i^2}$ and $c'(\tau_i)\tau_i^{-1/2}-\frac{1}{2}c(\tau_i)\tau_i^{-3/2}$, and subsequently taking the limit $\tau_i\downarrow 0$ of $\frac{d\pi(\tau_i,N_i,\beta_i)}{d\tau_i}$ yields

$$
\begin{aligned}
&\lim_{\tau_i\downarrow 0}\frac{d\pi(\tau_i,N_i,\beta_i)}{d\tau_i}\\
&=\lim_{\tau_i\downarrow 0}\{c'(\tau_i)N_i\beta_i\}+(rT+L(1+hT))N_i\beta_i\lim_{\tau_i\downarrow 0}\left\{\frac{\tau_i c'(\tau_i)-c(\tau_i)}{\tau_i^2}\right\}\\
&\quad+\beta_i bT\phi\left(\Phi^{-1}\left(\frac{bT-1-hT)}{bT}\right)\right)\sqrt{\alpha N_i L}\lim_{\tau_i\downarrow 0}\left\{\frac{\tau_i c'(\tau_i)-\frac{1}{2}c(\tau_i)}{\tau_i\sqrt{\tau_i}}\right\}.
\end{aligned}
$$

We have that $\lim_{\tau_i\downarrow 0}\left\{\frac{\tau_i c'(\tau_i)-c(\tau_i)}{\tau_i^2}\right\}=-\infty$ and $\lim_{\tau_i\downarrow 0}\left\{\frac{\tau_i c'(\tau_i)-\frac{1}{2}c(\tau_i)}{\tau_i\sqrt{\tau_i}}\right\}=-\infty$, because $0\leq\lim_{\tau_i\downarrow 0}c(\tau_i)<\infty$ and $0\leq\lim_{\tau_i\downarrow 0}c'(\tau_i)<\infty$. Hence, $\lim_{\tau_i\downarrow 0}\frac{d\pi(\tau_i,N_i,\beta_i)}{d\tau_i}=-\infty$, and $\pi(\tau_i,N_i,\beta_i)$ is strictly decreasing as $\tau_i\downarrow 0$.

(ii) $\pi(\tau_i, N_i, \beta_i)$ is strictly increasing as τ_i tends to $\overline{\tau}$. Consider

$$\begin{aligned}
&\lim_{\tau_i \to \overline{\tau}} \frac{d\pi(\tau_i, N_i, \beta_i)}{d\tau_i} \\
&= \lim_{\tau_i \to \overline{\tau}} \{c'(\tau_i) N_i \beta_i\} + (rT + L(1+hT)) N_i \beta_i \lim_{\tau_i \to \overline{\tau}} \left\{ \frac{\tau_i c'(\tau_i) - c(\tau_i)}{\tau_i^2} \right\} \\
&\quad + \beta_i bT \phi \left(\Phi^{-1} \left(\frac{bT - 1 - hT)}{bT} \right) \right) \sqrt{\alpha N_i L} \lim_{\tau_i \to \overline{\tau}} \left\{ \frac{\tau_i c'(\tau_i) - \frac{1}{2} c(\tau_i)}{\tau_i \sqrt{\tau_i}} \right\}.
\end{aligned}$$

We know that $\lim_{\tau_i \to \overline{\tau}} \{c'(\tau_i) N_i \beta_i\} = \infty$ by the convexity of $c(\tau)$ and as $\lim_{\tau \to \overline{\tau}} c(\tau) = \infty$. Furthermore,

$$\begin{aligned}
\lim_{\tau_i \to \overline{\tau}} \left\{ \frac{\tau_i c'(\tau_i) - c(\tau_i)}{\tau_i^2} \right\} &= \lim_{\tau_i \to \overline{\tau}} \left\{ \frac{1}{\tau_i} \left(c'(\tau_i) - \frac{c(\tau_i)}{\tau_i} \right) \right\} \\
&> \lim_{\tau_i \to \overline{\tau}} \left\{ \frac{1}{\tau_i} \left(\frac{c(\tau_i)}{\tau_i} - \frac{\lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)]}{\tau_i} - \frac{c(\tau_i)}{\tau_i} \right) \right\} \\
&= \lim_{\tau_i \to \overline{\tau}} \left\{ -\frac{1}{\tau_i^2} \lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)] \right\} > -\infty.
\end{aligned}$$

The first inequality follows from the strict convexity of $c(\tau_i)$, which implies $c'(\tau_i) > \frac{c(\tau_i) - \lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)]}{\tau_i}$. The final equality holds because $\lim_{\tilde{\tau}_i \downarrow 0} c(\tilde{\tau}_i) < \infty$ (by assumption) and $\lim_{\tau \to \overline{\tau}} \left\{ -\frac{1}{\tau_i^2} \right\} > -\infty$. Similarly, we obtain

$$\begin{aligned}
\lim_{\tau_i \to \overline{\tau}} \left\{ \frac{\tau_i c'(\tau_i) - \frac{1}{2} c(\tau_i)}{\tau_i \sqrt{\tau_i}} \right\} &> \lim_{\tau_i \to \overline{\tau}} \left\{ \frac{\frac{1}{2} c(\tau_i) - \lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)]}{\tau_i \sqrt{\tau_i}} \right\} \\
&> \lim_{\tau_i \to \overline{\tau}} \left\{ -\frac{1}{\tau_i \sqrt{\tau_i}} \lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)] \right\} > -\infty,
\end{aligned}$$

where the first inequality follows from the strict convexity of $c(\tau_i)$, i.e., $c'(\tau_i) > \frac{c(\tau_i) - \lim_{\tilde{\tau}_i \downarrow 0} [c(\tilde{\tau}_i)]}{\tau_i}$. The second inequality follows from $c(\tau_i) > 0$ for all $\tau_i \in (0, \overline{\tau})$. Hence, we have $\lim_{\tau_i \to \overline{\tau}} \frac{d\pi(\tau_i, N_i, \beta_i)}{d\tau_i} = \infty$ and $\pi(\tau_i, N_i, \beta_i)$ is strictly increasing as $\tau_i \to \overline{\tau}$. By combining *(i)* and *(ii)* with the strict convexity of $\pi(\tau_i, N_i, \beta_i)$, we obtain the desired result.

2.A.5 Proof of Lemma 2.5

Let $\overline{\tau} \in \mathbb{R}_+ \cup \{\infty\}$ and $f(\tau) = c(\tau) + \frac{r c(\tau) T}{\tau}$. Then, $f(\tau)$ is twice differentiable and strictly convex by definition of $c(\tau)$ and Assumption 2.2. Furthermore, we have that $\lim_{\tau \downarrow 0} f(\tau) = \infty$ as $\lim_{\tau \downarrow 0} c(\tau) < \infty$. We also have $\lim_{\tau \to \overline{\tau}} f(\tau) = \infty$, because $\lim_{\tau \to \overline{\tau}} c(\tau) = \infty$ and $0 < \lim_{\tau \to \overline{\tau}} 1/\tau < \infty$. Hence, there exists a unique, positive, finite minimizer $\hat{\tau}^*$ of $f(\tau)$.

Next, for each $i \in I$ the cost expression is $\hat{\pi}(\tau_i, N_i, \beta_i) = \beta_i N_i \left(c(\tau_i) + \frac{r c(\tau_i) T}{\tau_i} \right)$. The

minimum of $\hat{\pi}(\tau_i, N_i, \beta_i)$ over τ_i is solely determined by $c(\tau_i) + \frac{rc(\tau_i)T}{\tau_i} = f(\tau_i)$ for which the optimum is $\hat{\tau}^*$. Hence, $\hat{\tau}^* = \hat{\tau}_i^*$.

2.A.6 Proof of Corollary 2.1

($\Leftarrow$) Let $\beta_q \leq \sum_{i \in J} \frac{N_i \beta_i}{N_q}$. Then, we have for the costs of the common component:

$$\begin{aligned}\hat{\pi}(\hat{\tau}_q^*, N_q, \beta_q) &\leq \sum_{i \in J} \frac{N_i \beta_i}{N_q} N_q c(\hat{\tau}_q^*) \left(1 + \frac{rT}{\hat{\tau}_q^*}\right) \\ &= \sum_{i \in J} \frac{N_i \beta_i}{N_q} N_q c(\hat{\tau}_i^*) \left(1 + \frac{rT}{\hat{\tau}_i^*}\right) \\ &= \sum_{i \in J} \hat{\pi}(\hat{\tau}_i^*, N_i, \beta_i),\end{aligned}$$

where the inequality follows from inserting $\beta_q \leq \sum_{i \in J} \frac{N_i \beta_i}{N_q}$, and the first equality follows from $\hat{\tau}_i^* = \hat{\tau}_j^*$ for $i, j \in I$.

($\Rightarrow$) Let $\hat{\pi}(\hat{\tau}_q^*, N_q, \beta_q) \leq \sum_{i \in J} \hat{\pi}(\hat{\tau}_i^*, N_i, \beta_i)$. We have,

$$\begin{aligned}\sum_{i \in J} \hat{\pi}(\hat{\tau}_i^*, N_i, \beta_i) &= \sum_{i \in J} \beta_i N_i c(\hat{\tau}_i^*) \left(1 + \frac{rT}{\hat{\tau}_i^*}\right) \\ &= \sum_{i \in J} \beta_i N_i c(\hat{\tau}_q^*) \left(1 + \frac{rT}{\hat{\tau}_q^*}\right) \\ &\geq \beta_q N_q c(\hat{\tau}_q^*) \left(1 + \frac{rT}{\hat{\tau}_q^*}\right),\end{aligned}$$

where second equality follows because $\hat{\tau}_i^* = \hat{\tau}_j^*$ for components $i, j \in I$. The inequality follows by assumption. Rewriting the above yields $\beta_q \leq \sum_{i \in J} \frac{N_i \beta_i}{N_q}$.

This implies that $\sum_{i \in J} \frac{\beta_i N_i}{N_q}$ is the switching curve that determines whether the non–anticipating approach selects commonality. Thus, $\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta}) = \sum_{i \in J} \frac{\beta_i N_i}{N_q}$.

2.A.7 Proof of Proposition 2.1

The denominator of $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ is constant as $\sum_{i \in J} N_i = N_q$ is constant for any $N_q \in \mathbb{N}$.

(a) Let us recall the numerator of $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$:

$$\begin{aligned}&\sum_{i \in J} \beta_i c(\tau_q^*) \left(1 + \frac{rT + L(1 + hT)}{\tau_q^*}\right) N_i \\ &+ \sum_{i \in J} \beta_i c(\tau_q^*) bT \sqrt{\frac{\alpha N_i L}{\tau_q^*}} \phi\left(\Phi^{-1}\left(\frac{bT - 1 - hT}{bT}\right)\right).\end{aligned}$$

From the above, we see that the first summation of the numerator is constant, as $\beta_i = \beta_j,\ \forall i, j \in J$. Hence, we focus on the numerator's second summation. Suppose that $\boldsymbol{N}$ is such that there exists $j, k \in J$ satisfying $N_j - N_k > 1$, then a swap from one unit of j to k increases the second term of the numerator, and therefore we find that $\tilde{\Theta}(\boldsymbol{N} - e_j + e_k, \boldsymbol{\beta}) \geq \tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$, where e_j is the unit vector with value 1 at j and 0 elsewhere. Indeed, note that

$$\begin{aligned}
&\sqrt{N_j - 1} + \sqrt{N_k + 1} + \sum_{i \in J \setminus \{j,k\}} \sqrt{N_i} - \sum_{i \in J} \sqrt{N_i} + \sum_{i \in J} \sqrt{N_i} \\
&\quad = \sqrt{N_j - 1} + \sqrt{N_k + 1} - \sqrt{N_j} - \sqrt{N_k} + \sum_{i \in J} \sqrt{N_i} \\
&\quad > \sqrt{N_k} + \sqrt{N_k + 1} - \sqrt{N_k + 1} - \sqrt{N_k} + \sum_{i \in J} \sqrt{N_i} = \sum_{i \in J} \sqrt{N_i},
\end{aligned}$$

where the inequality follows from the assumption that $N_j - N_k > 1$.

(b) To prove the assertion, we relax the integrality of N_i. Furthermore, let us define $A = c(\tau_q^*)\left(1 + \frac{rT + L(1+hT)}{\tau_q^*}\right)$, $B = c(\tau_q^*) bT \sqrt{\frac{\alpha L}{\tau_q^*}} \phi\left(\Phi^{-1}\left(\frac{bT - 1 - hT}{bT}\right)\right)$, and $N_q = \sum_{i \in J} N_i$. As the denominator of $\tilde{\Theta}(\boldsymbol{N}, \boldsymbol{\beta})$ is constant, we are interested in maximizing the numerator:

$$\boldsymbol{N^*} = \underset{\boldsymbol{N}}{\operatorname{argmax}} \left\{ A \sum_{i \in J} N_i \beta_i + B \sum_{i \in J} \sqrt{N_i} \beta_i \ : \ \sum_{i \in J} N_i = N_q, N_i \geq 0 \right\},$$

which is equivalent to

$$(\boldsymbol{N^*}, \boldsymbol{v}) = \underset{\boldsymbol{N}, \boldsymbol{v}}{\operatorname{argmax}} \left\{ A \sum_{i \in J} N_i \beta_i + B \sum_{i \in J} \sqrt{N_i} \beta_i \ : \ \sum_{i \in J} N_i = N_q, N_i - v_i^2 = 0 \right\},$$

where $\boldsymbol{N}$ and $\boldsymbol{v}$ are the vectors of all N_i and v_i, $i \in J$. We square v_i to enforce non–negativity for any value of $v_i \in \mathbb{R}$. Consequently, the Lagrangian of the above problem is

$$\mathcal{L}(\boldsymbol{N}, \boldsymbol{v}, \lambda, \boldsymbol{\mu}) = A \sum_{i \in J} N_i \beta_i + B \sum_{i \in J} \sqrt{N_i} \beta_i - \sum_{i \in J} \lambda (N_i - N_q) - \sum_{i \in J} \mu_i (N_i - v_i^2),$$

with $\boldsymbol{\mu}$ denoting the vector of all μ_i, $i \in J$. The Lagrange multipliers are λ and $\boldsymbol{\mu}$. The first order conditions, required to maximize the Lagrangian, are given by:

$$\frac{\partial \mathcal{L}}{\partial N_i} = A\beta_i + \frac{\beta_i B}{2\sqrt{N_i}} - \lambda - \mu_i = 0, \quad \forall i \in J \tag{2.5}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_i} = N_i - v_i^2 = 0, \quad \forall i \in J \tag{2.6}$$

$$\frac{\partial \mathcal{L}}{\partial v_i} = 2\mu_i v_i = 0, \quad \forall i \in J \tag{2.7}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{i \in J} N_i - N_q \qquad = 0. \tag{2.8}$$

$v_i = 0$ cannot occur, as Eq. (2.6) implies $N_i = 0$, which violates feasibility in Eq. (2.5). Thus, we have that $v_i > 0$ or $v_i < 0$. From Eq. (2.7) we have that $\mu_i = 0$, and from Eq. (2.6) we know that $N_i = v_i^2$. We use Eq. (2.5) to determine the optimal size of the installed base:

$$\sqrt{N_i^*} = \sqrt{v_i^{*2}} = \frac{B}{2} \frac{\beta_i}{\lambda - A\beta_i}.$$

Since we square v_i^*, we have $\sqrt{v_i^{*2}} > 0$ for $v_i > 0$ and for $v_i < 0$, implying that $\lambda > A\beta_i$, because $A, B, \beta_i \geq 0$. The three cases for v_i show that $\mu_i = 0$ and $\lambda > A\beta_i$, for all $i \in J$ must hold in order to have a feasible solution. Thus, we have for the optimal installed base size:

$$N_i^* = \frac{B^2}{2} \left[\frac{\beta_i}{\lambda - A\beta_i} \right]^2.$$

Since $\lambda > A\beta_i$, for all $i \in J$, we have that N_i^* increases with increasing β_i. Therefore, if $\beta_i \leq \beta_j$ then $N_i^* \leq N_j^*$ for all $j \in J$.

2.B. Poisson distributed demand

We explore how to solve our model if the demand during L is Poisson distributed rather than normally distributed. We only consider the anticipating approach in this appendix, because the non–anticipating approach can be derived easily from the results that follow. Let us denote the LCC of component $i \in I$, under Poisson distributed demand during L, by

$$\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i) = \beta_i c(\tau_i)(N_i + s_i) + h s_i T \beta_i c(\tau_i) + r \beta_i c(\tau_i) \frac{N_i T}{\tau_i} + bT\mathbb{E}[(D_i^p(L, N_i, \tau_i) - s_i)^+],$$

where $D_i^p(L, N_i, \tau_i)$ is the Poisson distributed demand. We are interested in the optimal reliability level τ_i^p and turnaround stock level s_i^p for Poisson distributed that minimize $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$. We have

Lemma 2.6 *For a given $s_i \in \mathbb{N}$, $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ is convex in τ_i, and $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ is minimized by a positive, unique, finite $\tau_i^p : s_i \in \mathbb{N}$. This $\tau_i^p : s_i \in \mathbb{N}$ is the solution to the first order condition.*

Proof. Let $s_i \in \mathbb{N}$ and $\tau_i \in (0, \overline{\tau})$. Then, all but the last term in $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ are convex in τ_i by Lemma 2.4 and because $c(\tau)$ satisfies Assumption 2.2. Next, we show

that the expected downtime $\mathbb{E}[(D_i^p(L,N_i,\tau_i)-s_i)^+]$ is also convex in τ_i for a given $s_i \in \mathbb{N}$. The first and second order derivative of the expected downtime is negative and positive, respectively.

$$\begin{aligned}
&\frac{d\mathbb{E}[(D_i^p(L,N_i,\tau_i)-s_i)^+]}{d\tau_i} \\
&= -\sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \left(x\frac{N_iL}{\tau_i^2}\left(\frac{N_iL}{\tau_i}\right)^{x-1}\right) - \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \frac{N_iL}{\tau_i^2}\left(\frac{N_iL}{\tau_i}\right)^{x} \\
&= -\sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \frac{x}{\tau_i}\left(\frac{N_iL}{\tau_i}\right)^{x} - \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \frac{1}{\tau_i}\left(\frac{N_iL}{\tau_i}\right)^{x+1} \\
&< 0, \qquad (2.9)
\end{aligned}$$

$$\begin{aligned}
&\frac{d^2\mathbb{E}[(D_i^p(L,N_i,\tau_i)-s_i)^+]}{d^2\tau_i} \\
&= \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \left[\left(\frac{N_iL}{\tau_i}\right)^{x} \frac{x(1+x)}{\tau_i^2}\right] + \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \frac{x}{\tau_i^2}\left(\frac{N_iL}{\tau_i}\right)^{x+1} \\
&+ \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \left(\frac{N_iL}{\tau_i}\right)^{x+1}\left(\frac{N_iL+\tau_i}{\tau_i^3}\right) \\
&+ \sum_{x=s_i+1}^{\infty} \frac{x-s_i}{x!} e^{-\frac{N_iL}{\tau_i}} \left(\frac{N_iL}{\tau_i}\right)^{x+1} \frac{x+1}{\tau_i^2} \\
&> 0.
\end{aligned}$$

Hence, $\mathbb{E}\left[(D_i^p(L,N_i,\tau_i)-s_i)^+\right]$ is convex and decreasing in τ_i for a given s_i. Thus $\tilde{\pi}^p(\tau_i,s_i,N_i,\beta_i)$ is convex.

Next, we show that $\tilde{\pi}^p(\tau_i,s_i,N_i,\beta_i)$ is strictly decreasing for $\tau_i \downarrow 0$. We have that

$$\begin{aligned}
0 <& \lim_{\tau_i\downarrow 0} \frac{d}{d\tau_i}\beta_i c(\tau_i)(N_i+s_i) < \infty, \\
0 <& \lim_{\tau_i\downarrow 0} \frac{d}{d\tau_i} h s_i T \beta_i c(\tau_i) < \infty, \\
& \lim_{\tau_i\downarrow 0} \frac{d}{d\tau_i} r\beta_i c(\tau_i) N_i T/\tau_i = -\infty, \\
& \lim_{\tau_i\downarrow 0} \left\{ \frac{d\mathbb{E}[(D_i^p(L,N_i,\tau_i)-s_i)^+]}{d\tau_i} \right\} < 0.
\end{aligned}$$

The first three expressions are derived similar to the the proof of Lemma 2.4. The last expression holds by Eq. (2.9). Hence, $\tilde{\pi}^p(\tau_i,s_i,N_i,\beta_i)$ is decreasing as $\tau_i \downarrow 0$.

Finally, we show that $\tilde{\pi}^p(\tau_i,s_i,N_i,\beta_i)$ is increasing as $\tau_i \to \overline{\tau}$. We have

$$\lim_{\tau_i\to\overline{\tau}} \frac{d}{d\tau_i}\beta_i c(\tau_i)(N_i+s_i) = \infty,$$

$$\lim_{\tau_i \to \overline{\tau}} \frac{d}{d\tau_i} h s_i T \beta_i c(\tau_i) = \infty,$$
$$\lim_{\tau_i \to \overline{\tau}} \frac{d}{d\tau_i} r \beta_i c(\tau_i) N_i T / \tau_i > -\infty$$
$$\lim_{\tau_i \to \overline{\tau}} \left\{ \frac{d\mathbb{E}[(D_i^p(L, N_i, \tau_i) - s_i)^+]}{d\tau_i} \right\} > -\infty.$$

The first three expressions are derived similar to the proof of Lemma 2.4. The last expression follows from Eq. (2.9), for which we have the finite limits $\lim_{\tau_i \to \overline{\tau}} e^{-\frac{N_i L}{\tau_i}}$, $\lim_{\tau_i \to \overline{\tau}} \left\{ \frac{x}{\tau_i} \right\}$, and $\lim_{\tau_i \to \overline{\tau}} \left\{ \left(\frac{N_i L}{\tau_i} \right)^{x+1} \right\}$ with $x \in \mathbb{N}$. Therefore, $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ is increasing as $\tau_i \to \overline{\tau}$.

Combining all the above yields that $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ is convex and there exists a positive, unique and finite τ_i^p that minimizes $\tilde{\pi}^p(\tau_i, s_i, N_i, \beta_i)$ given $s_i \in \mathbb{N}$. □

Lemma 2.6 implies that we can efficiently determine $\tau_i^p : s_i \in \mathbb{N}$ for each component $i \in I$. Next, we consider the commonality decision under Poisson demand during L. We let (τ_i^p, s_i^p) be the tuple corresponding to the optimal decision in terms of reliability and turnaround stock level. We are interested in the switching curve for β_q under Poisson demand. This switching curve is characterized by

$$\Theta^p(\boldsymbol{N}, \boldsymbol{\beta}) = \max \left\{ \beta_q \ : \ \tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) \leq \sum_{i \in J} \tilde{\pi}^p(\tau_i^p, s_i^p, N_i, \beta_i) \right\}.$$

The following result enables us to determine $\Theta^p(\boldsymbol{N}, \boldsymbol{\beta})$ efficiently.

Lemma 2.7 *$\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q)$ is monotone increasing in β_q, and there exists a positive, finite, unique value for β_q such that $\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) = \sum_{i \in J} \tilde{\pi}^p(\tau_i^p, s_i^p, N_i, \beta_i)$, which we denote by $\Theta^p(\boldsymbol{N}, \boldsymbol{\beta})$.*

Proof. Let $\beta_q > 0$ and $\tilde{\beta}_q > \beta_q$. We denote τ_q^p and $\tilde{\tau}_q^p$ as the optimal reliability levels of the common component under β_q and $\tilde{\beta}_q$, respectively. Similarly, we let s_q^p and $\tilde{s}_q^p$ denote the optimal turnaround stock levels of the common component under β_q and $\tilde{\beta}_q$, respectively. Then, $\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) \leq \tilde{\pi}^p(\tilde{\tau}_q^p, \tilde{s}_q^p, N_q, \beta_q) < \tilde{\pi}^p(\tilde{\tau}_q^p, \tilde{s}_q^p, N_q, \tilde{\beta}_q)$, where the first inequality follows by the suboptimality of $(\tilde{\tau}_q^p, \tilde{s}_q^p)$ in $\tilde{\pi}^p(\tilde{\tau}_q^p, \tilde{s}_q^p, N_q, \beta_q)$. The latter inequality results from the linear dependency of $\tilde{\pi}^p(\tau_q, s_q, N_q, \beta_q)$ on β_q. Thus $\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q)$ is monotone increasing in β_q.

Next, we show that there exists a positive, finite, and unique β_q such that $\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) = \sum_{i \in J} \tilde{\pi}^p(\tau_i^p, s_i^p, N_i, \beta_i)$. We have $\lim_{\beta_q \downarrow 0} \tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) = \lim_{\beta_q \downarrow 0} bT\mathbb{E}[(D_q^p(L, N_q, \tau_q^p) - s_q^p)^+] = 0$, where the final inequality follows because $s_q^p = \infty$ if $\beta_q \downarrow 0$. This implies that are no costs for producing s_i^p units and no storage costs; therefore, $s_q^p = \infty$. Furthermore, note that $\tau_q^p \to \overline{\tau}$ as there is no cost for increasing

the reliability if $\beta_q \downarrow 0$. Secondly, we have that $\lim_{\beta_q \to \infty} \tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) = \infty$, because τ_q^p is finite. Hence, there exists a positive, finite, and unique β_q such that $\tilde{\pi}^p(\tau_q^p, s_q^p, N_q, \beta_q) = \sum_{i \in J} \tilde{\pi}^p(\tau_i^p, s_i^p, N_i, \beta_i)$. □

Hence, for the anticipating approach, we are able to determine the optimal reliability level for each component $i \in I$ under Poisson distributed demand during L by enumerating turnaround stock levels $s_i \in \mathbb{N}$ and subsequently using Lemma 2.6. Moreover, we can numerically determine the switching curve for commonality under the anticipating approach by Lemma 2.7. This differs with normal distributed demand, because the latter allows for an analytical characterization of the switching curve. Finally, we note that a similar analysis can be done for the non–anticipating approach by using the same steps as above.

2.C. Extra numerical insights

2.C.1 Additional numerical reliability results

The more detailed numerical results for the relative reliability difference between the non–anticipating and anticipating approach are given in Tables 2.3 and 2.4. The terms average, minimum, and maximum are abbreviated by avg, min, and max, respectively.

The results indicate that the relative reliability differences are large and can be as much as 27%. We observe that the costs of storing service parts h, the cost of a repair r, and the horizon T have a particularly large effect on the differences in the optimal reliability levels between both approaches. When the storage costs of service parts h increase, the effect of considering service parts becomes larger, and therefore we observe larger differences in the optimal reliability levels between the two approaches. If the repair costs per repair increase the difference between both approaches reduces because the repair element becomes more dominant in the life cycle cost expression of each approach. Finally, if T increases, the reliability difference decreases, because T has a large effect on the repair costs. Consequently, the repair costs become a strong element in the LCC expression of both approaches. Thus, the reliability difference decreases.

Overall, the average differences between the optimal reliability levels are substantial and in the order of magnitude of 10%. Hence, neglecting service parts from the reliability decision underestimates the optimal reliability levels. The results in Tables 2.3 and 2.4 imply that the optimal reliability decisions can substantially differ when considering service parts, by as much as 27%. These differences directly influence the intensity at which systems fail and thus can have a major effect on the performance and costs of after–sales services, e.g. service part provisioning.

		τ_1			τ_2		
		avg (%)	min (%)	max (%)	avg (%)	min (%)	max (%)
h	0.015	8.65	2.91	19.62	8.57	2.91	19.62
	0.03	12.58	4.44	27.75	12.50	4.44	27.75
r	0.2	13.25	5.01	27.75	13.15	5.01	27.75
	0.3	7.98	2.91	17.61	7.92	2.91	17.61
b	100,000	10.51	2.91	27.20	10.43	2.91	27.20
	1,000,000	10.72	2.99	27.75	10.64	2.99	27.75
T	180	13.14	4.78	27.75	13.03	4.78	27.75
	360	8.10	2.91	17.52	8.05	2.91	17.52
L	3	9.82	2.91	24.31	9.74	2.91	24.31
	4	11.41	3.42	27.75	11.34	3.42	27.75
p_1	500	11.75	3.79	27.75	11.67	3.79	27.75
	5,000	9.48	2.91	24.02	9.40	2.91	24.02
p_2	100	9.48	2.91	24.02	9.40	2.91	24.02
	1,000	11.75	3.79	27.75	11.67	3.79	27.75
k	1	11.05	2.91	27.75	10.96	2.91	27.75
	2	10.18	3.02	24.09	10.11	3.02	24.09
N_1	100	11.44	3.63	27.75	9.89	2.91	24.31
	200	10.41	3.15	25.46	10.41	3.15	25.46
	300	9.89	2.91	24.31	11.44	3.63	27.75

Table 2.3: Reliability differences for the dedicated components of Testbed 2.1

2.C.2 Additional numerical commonality results

Next, we study the numerical results for the commonality decision; see Table 2.5. We abbreviate the terms average, minimum, and maximum by avg, min, and max, respectively. The results indicate that the difference in the commonality decision is mainly affected by the storage costs h, the horizon T, and by the installed base size N_1. Higher storage costs h and a longer horizon T make it more important to make the right commonality decision, because service part aspects are more important for the LCC (higher storage costs and longer usage of service parts). The sizes of the installed bases also have a large influence on the difference in the commonality decision between both approaches. The installed base sizes determine how attractive the pooling of service parts is, which is exploited under commonality. Hence, we see that N_1 determines the relative commonality difference between both approaches. The other parameters play smaller roles in determining $\left((\Theta(\boldsymbol{N},\boldsymbol{\beta})/\hat{\Theta}(\boldsymbol{N},\boldsymbol{\beta})-1\right)\times 100\%$.

2.C.3 Additional numerical LCC results

We use Testbed 2.1 also to study the relative LCC differences $\Delta\pi(\boldsymbol{N},\boldsymbol{\beta},\beta_q)$. The

		τ_q		
		avg (%)	min (%)	max (%)
h	0.015	7.65	2.76	15.93
	0.03	11.48	4.26	23.58
r	0.2	11.92	4.76	23.58
	0.3	7.21	2.76	15.00
b	100,000	9.49	2.76	23.22
	1,000,000	9.65	2.84	23.58
T	180	11.66	4.51	23.58
	360	7.47	2.76	15.98
L	3	8.75	2.76	19.96
	4	10.38	3.27	23.58
p_1	500	10.72	3.83	23.58
	5,000	8.41	2.76	19.83
p_2	100	8.41	2.76	19.83
	1,000	10.72	3.83	23.58
k	1	9.84	2.76	23.58
	2	9.30	2.88	21.52
N_1	100	9.57	2.76	23.58
	200	9.57	2.76	23.58
	300	9.57	2.76	23.58

Table 2.4: Reliability differences for the common component of Testbed 2.1

results are presented in Table 2.6, with the abbreviations avg (average), min (minimum), and max (maximum). The results indicate that the most important parameters for $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ are the service part storage costs h and the cost of a repair r, in addition to the installed base N_1 and the relative unit cost factors β_2 and β_q. The cost difference between both approaches increases when the storage costs of service parts h increases. This is a result from amplifying the service part costs in the LCC. Similarly, the cost difference between both approaches decreases as r increases, because the repair cost play a larger role in the LCC of both approaches, and thus the difference between both approaches decreases.

Lastly, we consider the sizes of the installed bases (which follow from N_1) and the relative unit cost factors β_2 and β_q in Table 2.6. All of these parameters interact with one another resulting in the relative LCC differences from Table 2.6. This interaction is apparent from Figure 2.4 for Example 2.1. It is difficult to conclude *how* the installed bases and the relative unit cost factors influence $\Delta\pi(\boldsymbol{N}, \boldsymbol{\beta}, \beta_q)$ individually, but we do see that their effect can be substantial.

		avg (%)	min (%)	max (%)
h	0.015	4.25	2.34	6.42
	0.03	6.57	3.68	9.59
r	0.2	5.64	2.46	9.59
	0.3	5.18	2.34	8.63
b	100,000	5.24	2.33	9.03
	1,000,000	5.58	2.50	9.59
T	180	4.58	2.33	7.43
	360	6.24	3.35	9.59
L	3	5.14	2.33	8.92
	4	5.68	2.64	9.59
p_1	500	5.58	2.62	9.59
	5,000	5.24	2.33	9.42
p_2	100	5.24	2.33	9.42
	1,000	5.58	2.62	9.59
k	1	5.28	2.33	9.46
	2	5.54	2.52	9.59
N_1	100	4.87	2.33	8.48
	200	5.84	2.96	9.59
	300	5.53	2.67	9.50
β_2	1	5.50	2.61	9.59
	1.05	5.39	2.56	9.59
	1.1	5.39	2.51	9.59
	1.15	5.40	2.46	9.59
	1.2	5.40	2.42	9.59
	1.25	5.41	2.37	9.59
	1.3	5.41	2.33	9.59

Table 2.5: Commonality differences $(\Theta(\boldsymbol{N}, \boldsymbol{\beta})/\hat{\Theta}(\boldsymbol{N}, \boldsymbol{\beta}) - 1) \times 100\%$ of Testbed 2.1

		avg (%)	min (%)	max (%)
h	0.015	0.48	0.07	6.81
	0.03	1.12	0.18	10.67
r	0.2	1.02	0.17	10.67
	0.3	0.58	0.07	9.11
b	100,000	0.77	0.07	10.09
	1,000,000	0.83	0.07	10.67
T	180	0.83	0.11	9.08
	360	0.77	0.07	10.67
L	3	0.69	0.07	9.64
	4	0.91	0.11	10.67
p_1	500	0.86	0.07	10.67
	5,000	0.74	0.08	10.43
p_2	100	0.73	0.08	10.43
	1,000	0.87	0.07	10.67
k	1	0.81	0.08	10.67
	2	0.79	0.07	10.46
N_1	100	0.74	0.07	8.91
	200	0.84	0.07	10.67
	300	0.82	0.07	10.25
β_2	1	0.61	0.07	6.25
	1.05	0.80	0.07	8.07
	1.1	0.82	0.07	10.67
	1.15	0.81	0.07	8.77
	1.2	0.94	0.07	10.67
	1.25	0.81	0.07	8.27
	1.3	0.75	0.07	8.71
β_q	1	0.46	0.07	6.25
	1.05	1.71	0.07	10.67
	1.1	1.52	0.07	10.67
	1.15	0.97	0.07	8.91
	1.2	0.84	0.07	7.64
	1.25	0.67	0.07	6.47
	1.3	0.53	0.07	2.38
	1.35	0.53	0.07	1.77
	1.4	0.53	0.07	1.77
	1.45	0.53	0.07	1.77
	1.5	0.53	0.07	1.77

Table 2.6: Relative LCC differences of Testbed 2.1

3

Design of disjoint Line Replaceable Units

3.1. Introduction

In this chapter, we study an OEM who is responsible for the maintenance of multiple systems. The OEM is negatively impacted by system failures, because a failure generates costs during the time the system is not functioning. As a result, the OEM tries to reduce the time needed to restore the system to a functioning state. This problem is amplified in industries where systems are critical for operations, e.g. the railway industry (Bombardier/Dutch Railways), the semiconductor industry (ASML), the trucking industry (PACCAR/Volkswagen Group), and the aviation industry (Pratt & Whitney). One of the mechanisms used to reduce the time needed to restore systems to a functioning state is to use Line Replaceable Units (LRUs).

A LRU is a collection of parts that is replaced entirely from the system when one of the parts in the LRU fails. For example, if a motherboard together with the graphics card is one LRU in a laptop, and the motherboard or the graphics card fails, the entire LRU is replaced. The main advantage of a LRU is that it can reduce the time and cost spent on replacement, because the grouping of parts may simplify the replacement actions, e.g. fewer screws need to be removed. On the other hand, failed LRUs are replaced by new ones, and thus we need to purchase new LRUs or repair the failed LRU. These LRUs typically become more expensive as they contain more parts. Hence, the OEM wants to design LRUs such that the cost trade–off between replacement costs and purchase costs (or repair costs) is minimized.

The objective of this chapter is to develop a mathematical decision support model that determines the optimal design of LRUs for a given system structure. Our contributions are the following. We present (i) a novel way of representing a system with multiple parts that are connected to each other, and we incorporate the disassembly sequences that exist for maintenance, e.g. we disconnect part A before part B. Next, we use our system description to define an optimization model – called LRU Design – that minimizes the sum of the replacement and purchase costs by optimizing the LRU designs. We present (ii) LRU Design's most natural formulation: a binary non–linear program (BNLP), which we transform into a binary linear program. Thirdly, (iii) we provide a set partitioning formulation of LRU Design that allows for branch–and–price algorithms. We prove (iv) that branching is unnecessary in solving this set partitioning formulation to optimality, i.e., using pure pricing algorithms suffices. Furthermore, (v) we numerically illustrate that the set partitioning formulation is efficient for large instances, and we study the effects of various parameters on the model's outcome.

3.1.1 Literature

Our problem relates to multi–component maintenance research with structural dependencies. Structural dependency between parts means that some parts have to be replaced or removed before we can replace the failed part. Practitioners are frequently faced with this type of dependency, but the academic field studying it "is wide open . . . and there have only been a few articles published on this topic" (Nicolai and Dekker, 2008). Early research in this area was done by Thomas (1986), who poses the question whether to replace the entire car, the engine or just the piston rings in case the piston rings need replacement. More recently, Parada Puig and Basten (2015) have formalized the question posed by Thomas (1986) and proposed a model that determines the optimal LRU design such that costs are minimized subject to an availability constraint. The cost expression in Parada Puig and Basten (2015) assumes that the total costs of replacing and purchasing a LRU are input to the model. However, there may exist many potential LRUs, and therefore, the approach by Parada Puig and Basten (2015) requires a large effort for inputting all the data. In our approach, the costs of replacing and purchasing a LRU are derived from the graphs that describe the structure existing between the various parts in a system. Therefore, we do not have to determine the costs for each possible LRU a priori. Furthermore, both Thomas (1986) and Parada Puig and Basten (2015) assume that the underlying system structure is a tree, e.g. a Bill of Materials (BOM). One of the issues with this assumption is that a system's structure is hardly a tree in practice; e.g. a motherboard is connected to a laptop's main frame and a battery is connected to the motherboard and main frame, and thus we cannot have a tree structured system. Our work overcomes this issue by allowing arbitrary system structures. Moreover, the tree structured approach restricts the number of possible LRUs, as only the nodes in the tree are potential LRUs.

Another line of related research studies LRUs (which are called modules) from a systems engineering perspective, where "a module (LRU) is a unit whose parts are powerfully connected among themselves, and relatively weakly connected to parts in other units" (Baldwin and Clark, 2000). This literature stream typically describes a system in terms of parts that are connected to each other, and these connections are commonly depicted in a Design Structure Matrix (Steward, 1981). However, this approach neglects the disassembly sequence that exist for the replacement of LRUs (or modules). Most research in this stream aims to define measures of modularity and to optimize these. Such measures typically focus on the connections between parts, and the measures prefer a high number of intra–LRU connections and a low number of inter–LRU connections; see Newcomb et al. (1998), Sharman and Yassine (2004), and Sosa et al. (2007).

Work on (dis)assembly sequencing also has similarities to our work, because this stream models the (dis)assembly sequence that exists between parts in much detail (De Fazio and Whitney, 1987; Gupta and Krishnan, 1998; Lambert, 2007). Research in this area optimizes the (dis)assembly sequence. We do not optimize this sequence, but we consider it to be given and focus on optimizing the design of LRUs.

Finally, our work also relates to several operations research studies that consider the impact of modular design on operations. These studies are often combinatorial in nature and aim to design product configurations such that the demand for end products is met and the total costs are minimized; see for example Swaminathan and Tayur (1998), Thonemann and Brandeau (2000), and Briant and Naddef (2004). The structure in their problems superficially resembles ours, because we also study configurations of parts, which are LRUs in our case. The main difference is that we model the connections between parts and the disassembly sequences that exist for maintenance, while research in this stream does not.

The rest of this chapter is outlined as follows. In Section 3.2, we discuss how we represent a general system that consists of multiple parts. We discuss how parts are connected to each other and we present a method of incorporating disassembly sequences in the system representation. Subsequently, we use this representation to present our optimization model LRU Design. We present the most natural formulation of LRU Design in Section 3.3: a binary non–linear programming (BNLP) formulation, which we linearize in order to obtain a binary linear program (BLP). In Section 3.4, we discuss a set partitioning formulation of LRU Design and we prove that the set partitioning formulation can be solved by pure pricing algorithms, i.e., branching is unnecessary. Finally, we numerically compare the computation times of the BLP formulation to computation times of the set partitioning formulation, and we illustrate the effects of various parameter perturbations on the model's outcome.

3.2. Model

We start by explaining how we represent a system consisting of multiple parts, various connections between parts, and multiple disassembly sequences that determine in what order connections must be broken. We do this, first, by means of an example. Subsequently, we generalize our system representation from the example, and we discuss the definition of a LRU design. Finally, we present our optimization model called LRU Design.

3.2.1 An illustrative example

Let us consider a situation in which we design and repair laptops. Our objective is to design the LRUs such that we minimize the total costs for the repair shop; i.e., to determine the optimal LRU design that trades off the costs of purchasing new LRUs and the costs spent on replacing LRUs.

Our example is based on data for a Dell Precision 7710 laptop (Dell Inc., 2016). Each part has a purchase cost and a failure rate (in failures per year). We list the estimated purchase cost[1] and fictitious failure rate for all parts (in failures per year) in Table 3.1.

Identifier	Part Name	Part Cost ($)	Failure rate (failures/year)
A	Battery	180	0.3
B	Hard Disk Drive	170	0.2
C	Keyboard	45	0.001
D	WLAN Card	50	0.15
E	Palm Rest	45	0.001
F	Speakers	14	0.05
G	Heat Sink	75	0.1
H	4 GB Video Card	250	0.1
I	Display Housing	40	0.001
J	Display Front Cover	20	0.001
K	Display Bezel	170	0.25
L	Motherboard	270	0.25
M	Computer Base	50	0.001

Table 3.1: Part identifier list

Each of the parts is connected to other parts, e.g. the Palm Rest is screwed to the Computer Base, the Palm Rest is wired to the Motherboard, and the Palm Rest is screwed to the Keyboard. Thus, there exist connections $\{E, M\}$, $\{E, L\}$, and $\{C, E\}$.

[1] The purchase cost of a part is its price that we found online on websites such as www.amazon.com.

In the event the Palm Rest fails and we wish to replace it individually, we have to break all the connections that the Palm Rest has with all other parts: $\{E, M\}$, $\{E, L\}$ and $\{C, E\}$. Breaking each connection takes a certain amount of time, which we can translate into costs by multiplying the time with a cost rate, e.g. the salary rate of the repair man. When the failed Palm Rest has been disconnected from the system, a new and identical Palm Rest from stock is installed into the system by reconnecting all the connections that have been broken previously (in order to remove the failed Palm Rest). This re–establishing of connections also costs time and can be translated into costs as well. Finally, a new Palm Rest is purchased to replenish the stock.

We can depict all parts, the connections, the failure rates, the purchase costs, and the costs of breaking and re–establishing connections in a weighted undirected graph. We let the parts correspond to vertices, and the part connections correspond to the edges. Furthermore, the failure rates and the purchase costs are attributes of the vertices, and the costs for breaking and re–establishing a connection correspond to the weight of an edge in the graph. So, for the laptop example, we derive such a graph by analyzing the Owner's manual (Dell Inc., 2016), and it is given in Figure 3.1. The costs of breaking and re–establishing a connection are estimates and are depicted on the edges.

We call the graph from Figure 3.1 the *connection graph*. The connection graph may suggest that we only need to break connections $\{E, L\}$, $\{E, M\}$ and $\{C, E\}$ in order to remove the Palm Rest. However, we know from the Owner's Manual that in order to disconnect the Palm Rest, we first have to break the connections that enable us to remove the Keyboard (C), the Hard Disk Drive (B), and the Battery (A); i.e., there exists a disassembly sequence. This implies that there exists a collection of connections that needs to be broken prior to breaking the connections $\{E, L\}$, $\{E, M\}$, or $\{C, E\}$ (Dell Inc., 2016). Consequently, we have a predecessor–successor relationship for breaking (and re–establishing) the connections depicted in Figure 3.1. We model such predecessor–successor relationships in a separate directed graph, which we call the *precedence graph*. An arc in the precedence graph from an edge $\{E, M\}$ to $\{E, L\}$ implies that connection $\{E, L\}$ has to be broken before connection $\{E, M\}$ can be broken. The precedence graph for the laptop (Dell Inc., 2016) is given in Figure 3.2.

If we combine the precedence graph (Figure 3.2) with the connection graph (Figure 3.1), we are able to list *all* connections that need to be broken for the replacement of an arbitrary part. For example, replacing the Palm Rest requires us to break $\{E, M\}$, $\{E, L\}$, and $\{C, E\}$ (see Figure 3.1), but for breaking connection $\{C, E\}$ we need to break the set of connections $\{\{A, L\}, \{A, M\}, \{B, M\}, \{B, L\}, \{C, L\}\}$ (see Figure 3.2). Similarly, we can determine all connections that need to be broken prior to $\{E, M\}$ and $\{E, L\}$. It appears that we have to break all connections $\{\{A, L\}, \{A, M\}, \{B, M\}, \{B, L\}, \{C, E\}, \{C, L\}, \{E, L\}, \{E, M\}\}$ in order to remove the Palm Rest (E). Analogously, we break all connections $\{\{A, L\}, \{A, M\}, \{B, M\}, \{B, L\}, \{C, E\}, \{C, L\}\}$ when the Keyboard (C) fails.

We can also decide to replace the Palm Rest (E) together with the Keyboard (C),

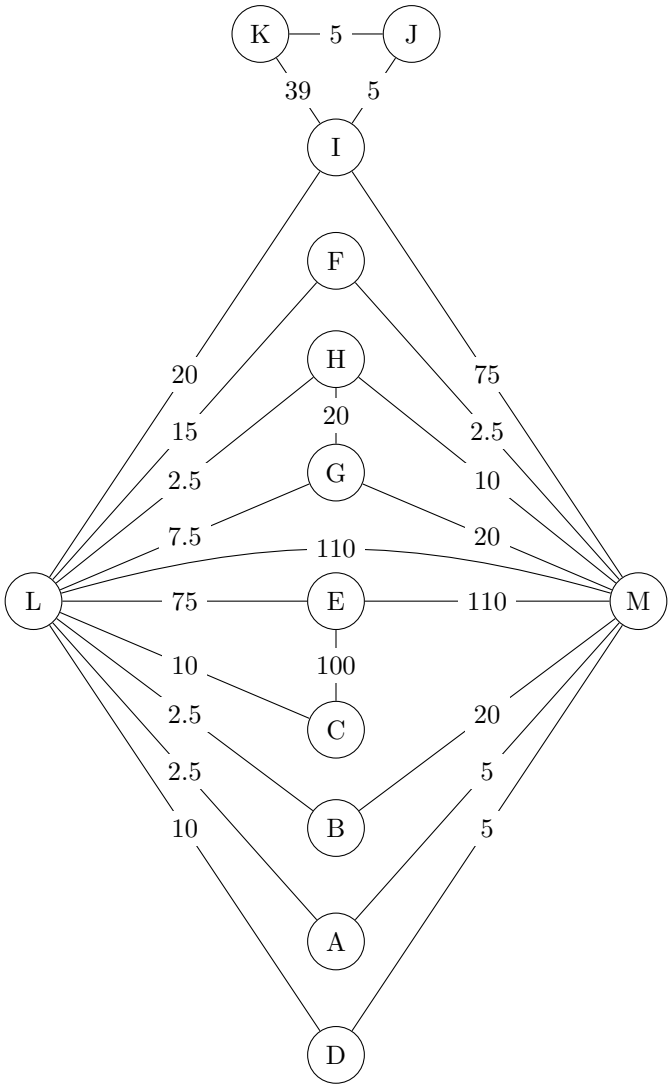


Figure 3.1: The laptop's connection graph

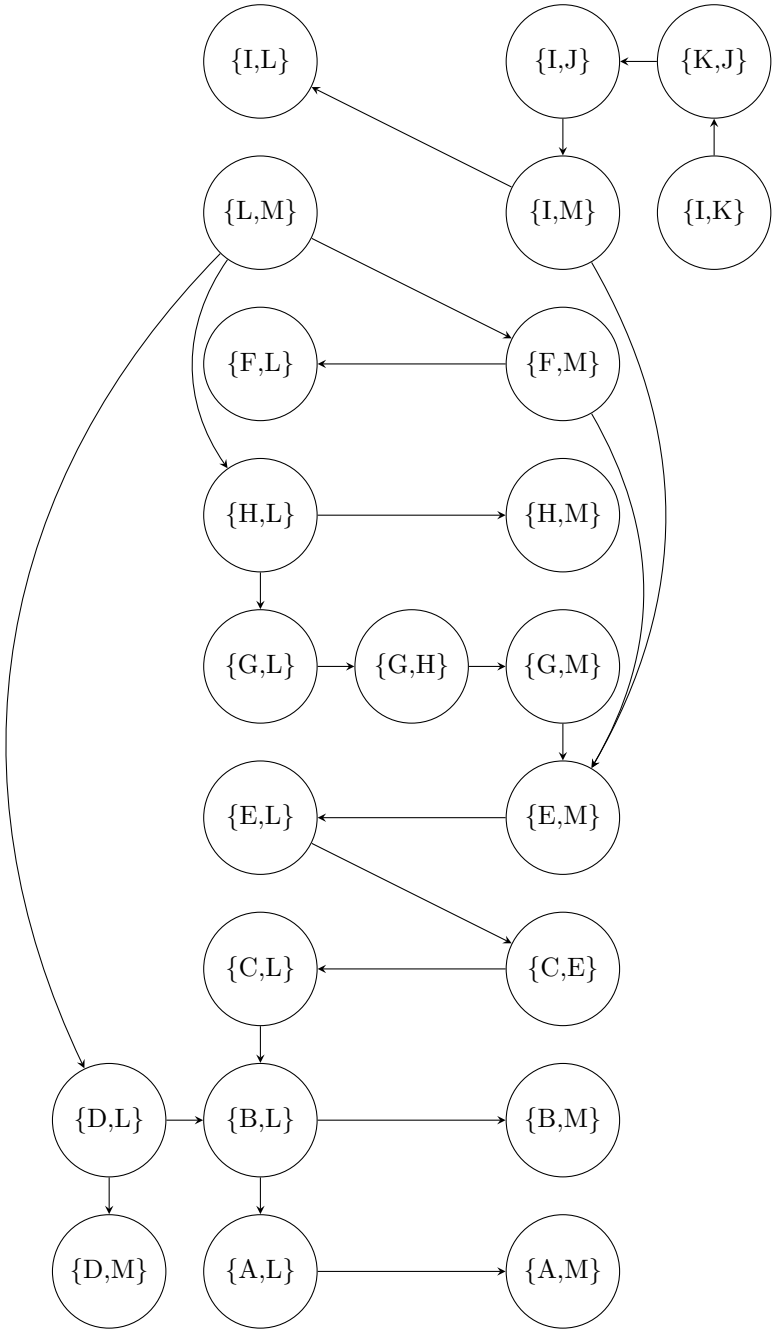


Figure 3.2: The laptop's precedence graph

i.e., define a LRU Q consisting of C and E. However, this implies that the engineer has to break all connections $\{\{A, L\}, \{A, M\}, \{B, M\}, \{B, L\}, \{C, L\}, \{C, E\}, \{E, L\}, \{E, M\}\}$ upon the failure of either the Palm Rest (E) or the Keyboard (C). As a consequence, we have to break the expensive edges $\{E, L\}$ and $\{E, M\}$ more often than when the Palm Rest and the Keyboard are separate LRUs. Furthermore, the LRU Q has a higher purchase costs as well as a higher failure rate compared to the Palm Rest and the Keyboard individually. So, it is better to keep the Palm Rest and the Keyboard as separate LRU instead of combining these two into one LRU Q. The repair shop is now interested in determining the optimal design of LRUs that minimizes the sum of the replacement and purchase costs, based on the connection graph in Figure 3.1 and the precedence graph in Figure 3.2.

3.2.2 A generic model

The example above illustrates how a system is built up and what relationships parts and connections have. Our approach used for the laptop also applies to more complicated systems such as a bogie in a train (Bombardier/Dutch Railways), a positioning module in a lithography system (ASML), a truck engine (PACCAR/Volkswagen Group), or a jet engine (Pratt & Whitney). The representation of such systems in terms of two graphs can be very useful to an OEM to enhance internal communication between departments; for instance to aid communication between the design department and the operations department. Moreover, the system representation can provide the engineers with a better insight in the system and its maintenance dependencies.

Let us consider a system that consists of multiple parts, and assume that maintenance is done upon the failure of a part. Moreover, we assume that we can accurately and instantaneously determine which part has failed, when the system fails as a whole. The system is defined by two graphs: a weighted undirected connection graph G and a directed precedence graph D. The graph $G = (V, E)$ is characterized by the set of vertices V and the set of edges E. The former set V corresponds to the parts in the system, and the latter set E are the connections between parts. Furthermore, each part in G has a failure rate $\lambda : V \to \mathbb{R}_+$ and purchase costs $\ell : V \to \mathbb{R}_+$, where $\mathbb{R}_+ = \{x \in \mathbb{R} \,|\, x > 0\}$. If a part fails, we break edges and breaking an edge costs $w : E \to \mathbb{R}_+$. We use the terms part and vertex interchangeably, as well as the terms connection and edge. At the end of this section, we discuss how we can use our model if there exist parts that are preventively replaced (Remark 3.1) and what happens when we repair parts (Remark 3.2).

Besides G, we have an unweighted acyclic directed graph $D = (E, A)$ that captures the disassembly sequences of the connections $e \in E$. The set A corresponds to the set of arcs, and an arc $(i, j) \in A$ from edge i to edge j exists if and only if edge j has to be broken *before* edge i can be broken. We assume that there exist only arcs between adjacent edges, i.e., all arcs in A satisfy $(\{u, v\}, \{v, x\}) \in A : u, v, x \in V$

and $u \neq v \neq x$. The graph D allows us to generate a set $H(e)$ of successor edges for each edge $e \in E$. This set $H(e)$ consists of all edges including the edge $e \in E$ that have to be disconnected in order to break e, and it can be determined by using the polynomial Algorithms 1 and 2 from Appendix 3.A. We remark that $H(e)$ is a directed tree rooted at $e \in E$.

Furthermore, we assume that G is connected, without loss of generality. If G is not connected, there do not exist arcs $(\{u, v\}, \{x, y\}) \in A$ such that u, v are in one connected component and x, y are in the other connected component. This follows because all arcs in A satisfy $(\{u, v\}, \{v, x\}) \in A : u, v, x \in V$ and $u \neq v \neq x$. Hence, if G were disconnected, we apply our model to each connected component of G with the precedence graph induced by the connected component.

Next, we define a LRU design as a partition S of the vertices V. Each LRU $Q \in S$ is characterized by the connections that need to be broken in order to replace the LRU Q from the system. We do so by defining the set $B(Q) = \{\{u, v\} \in E : u \in Q, v \in V \backslash Q\}$, which is the set of all edges that connect the LRU to the other parts of the system not in the LRU. That is, the set $B(Q)$ consists of edges that cross the LRU's boundary. For the removal of LRU Q, we do not only break all the edges $e \in B(Q)$, but also all the edges that need to be broken prior to breaking any edge $e \in B(Q)$. Hence, $\Gamma(Q) = \bigcup_{e \in B(Q)} H(e)$ is the set of edges that need to be broken in order to replace a LRU $Q \in S$.

In addition to the edges that need to be broken for the replacement of a LRU, each LRU Q is also characterized by its purchase costs and failure rate. We assume that we purchase an entirely new LRU if any of the parts in the LRU fails. Therefore, the purchase costs of a LRU equals the sum of the purchase cost of all parts in the LRU, i.e., the LRU's purchase costs are given by $\sum_{v \in Q} \ell(v)$. We relax this assumption in Remark 3.2. For the failure rate, we assume that a part $v \in V$ belongs to exactly one LRU, and each part $v \in Q$ triggers the replacement of the LRU Q. Hence, the total failure rate of a LRU $Q \in S$ is given by $\sum_{v \in Q} \lambda(v)$.

Next, we derive the cost expression for a LRU $Q \in S$. Upon the failure of LRU Q, we break all edges $e \in \Gamma(Q)$ resulting in the costs $\sum_{e \in \Gamma(Q)} w(e)$. Moreover, we replace the failed LRU by a new one that is purchased at costs $\sum_{v \in Q} \ell(v)$. Hence, we obtain the following average costs per time unit for LRU Q:

$$\omega(Q) = \sum_{e \in \Gamma(Q)} w(e) \sum_{v \in Q} \lambda(v) + \sum_{u \in Q} \ell(u) \sum_{v \in Q} \lambda(v). \tag{3.1}$$

Subsequently, as S is a partition of V and $Q \in S$, we obtain the following expression for the total costs per time unit of LRU design S:

$$\pi(S) = \sum_{Q \in S} \omega(Q) = \sum_{Q \in S} \sum_{e \in \Gamma(Q)} w(e) \sum_{v \in Q} \lambda(v) + \sum_{Q \in S} \sum_{u \in Q} \ell(u) \sum_{v \in Q} \lambda(v). \tag{3.2}$$

Given this definition of $\pi(S)$, we define the LRU Design problem as: What is the

LRU Design S that minimizes $\pi(S)$?

LRU Design has the property that each optimal solution S^* to LRU Design is such that each LRU $Q \in S^*$ is a connected subgraph of G.

Lemma 3.1 *Each LRU $Q \in S^*$ is a connected subgraph of G, for any optimal solution S^* to* LRU Design.

Proof. Let $\mathscr{J}$ be the finite set of connected components in the subgraph induced by a LRU Q. The set $\mathscr{J}$ is finite, because Q is finite, and $|\mathscr{J}| \geq 1$. The case of $|\mathscr{J}| = 1$ implies that Q is connected, which satisfies our claim. Thus, we consider $|\mathscr{J}| \geq 2$ in the remainder, and we have that $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ and for $\mathcal{J}_1, \mathcal{J}_2 \in \mathscr{J}$: $\mathcal{J}_1 \neq \mathcal{J}_2$. Furthermore, $\nexists\{u, v\} \in E : u \in \mathcal{J}_1, v \in \mathcal{J}_2$ with $\mathcal{J}_1, \mathcal{J}_2 \in \mathscr{J}$. This implies that $\Gamma(\mathcal{J}) \subset \Gamma(Q)$ for any $\mathcal{J} \in \mathscr{J}$ as $\mathcal{J} \subset Q$. Moreover, we have $\Gamma(Q) = \bigcup_{\mathcal{J} \in \mathscr{J}} \Gamma(\mathcal{J})$ as $\bigcup_{\mathcal{J} \in \mathscr{J}} \mathcal{J} = Q$ and $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ for $\mathcal{J}_1, \mathcal{J}_2 \in \mathscr{J}$: $\mathcal{J}_1 \neq \mathcal{J}_2$. In addition, $\sum_{\mathcal{J} \in \mathscr{J}} \sum_{v \in \mathcal{J}} \lambda(v) = \sum_{v \in Q} \lambda(v)$, because $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ with $\mathcal{J}_1, \mathcal{J}_2 \in \mathscr{J}$. Thus,

$$\begin{aligned}
\sum_{\mathcal{J} \in \mathscr{J}} \omega(\mathcal{J}) &= \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(\mathcal{J})} w(e) \sum_{v \in \mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in \mathcal{J}} \ell(u) \sum_{v \in \mathcal{J}} \lambda(v) \\
&\leq \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(Q)} w(e) \sum_{v \in \mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in \mathcal{J}} \ell(u) \sum_{v \in \mathcal{J}} \lambda(v) \\
&< \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(Q)} w(e) \sum_{v \in \mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in Q} \ell(u) \sum_{v \in \mathcal{J}} \lambda(v) \\
&= \sum_{e \in \Gamma(Q)} w(e) \sum_{v \in Q} \lambda(v) + \sum_{u \in Q} \ell(u) \sum_{v \in Q} \lambda(v) = \omega(Q),
\end{aligned}$$

where the first inequality follows from the fact that each $\Gamma(\mathcal{J}) \subset \Gamma(Q)$, $\forall \mathcal{J} \in \mathscr{J}$. The second inequality follows from the fact that $\mathcal{J} \subset Q$, $\forall \mathcal{J} \in \mathscr{J}$, and thus $\sum_{u \in \mathcal{J}} \ell(u) < \sum_{u \in Q} \ell(u)$, $\forall \mathcal{J} \in \mathscr{J}$. Hence, each LRU $Q \in S^*$ is a connected subgraph of G,, for any optimal solution S^* to LRU Design. □

We will use the result of Lemma 3.1 later throughout this chapter. We conclude this section by discussing two remarks that enable further generalization of LRU Design.

Remark 3.1 We assumed that all parts $v \in V$ are maintained upon failure. Now suppose some parts V_1 are maintained upon failure, while other parts V_2 are maintained preventively. If a part $v \in V_2$ fails in between two preventive maintenance actions, it is maintained upon its failure as well. The subsets of parts V_1 and V_2 are such that they partition all vertices V. Then, we interpret the failure rate $\lambda(v)$, $\forall v \in V$ as the replacement rate of a part. In case $v \in V_1$, this replacement rate corresponds to the failure rate (or equivalently, the frequency of corrective maintenance actions); and if $v \in V_2$, this replacement rate corresponds to the

maintenance frequency that results from the preventive and corrective maintenance actions.

Remark 3.2 We assume that we do not repair a LRU, and thus purchase a new one. If we relax this assumption and repair a failed part of a LRU offline, we incur total repair costs per time unit of $\sum_{v\in V} \lambda(v)q(v)$, where $q(v)$ are the repair costs of part v. However, if we repair a part of a LRU, we have to test the entire LRU to see whether it functions again. This means that we have to test each part in the LRU, and thus $\ell(v)$ now represents the costs of testing part $v \in V$ offline. Larger LRUs, now, have more parts that need to be tested before the LRU is certified as repaired. The total repair costs per time unit are sunk as $\sum_{v\in V} \lambda(v)q(v)$ is independent on the LRU design, but we still have the testing costs per time unit of LRU Q given by $\sum_{v\in Q} \lambda(v) \sum_{u\in Q} \ell(u)$. Hence, our model LRU Design still applies.

3.3. Binary programming formulation

In this section, we formulate LRU Design as a binary non–linear program (BNLP), and subsequently we linearize this BNLP to obtain a binary linear program (BLP). For the BNLP, we first relax the assumption that $\emptyset \notin S$. We let S' be a solution to LRU Design such that $\emptyset \in S'$. We have $S = \{Q \in S' : Q \neq \emptyset\}$ and S' satisfies $|S'| = |V|$. Next, we index each LRU in S' by $i \in \{1, \ldots, |V|\}$, i.e., we have LRUs $Q_i \in S'$ that are indexed by i. Furthermore, we create a binary variable y_{vi} that indicates whether a part $v \in V$ is assigned to LRU Q_i, $i \in \{1, \ldots, |V|\}$:

$$y_{vi} = \begin{cases} 1 & \text{if } v \in Q_i \\ 0 & \text{otherwise} \end{cases}, \quad \forall v \in V, \forall i \in \{1, \ldots, |V|\}.$$

We denote $\boldsymbol{Y}$ as the matrix consisting of all entries y_{vi}. Note that we can derive S' easily from $\boldsymbol{Y}$. We also define the auxiliary binary variable k_i^e that denotes whether edge $e \in E$ needs to be broken in order to replace LRU Q_i. We determine the value of k_i^e by considering all the edges $b \in B(Q_i)$. We determine $B(Q_i)$ by considering the edges $\{u, v\} \in E$ such that $y_{ui}(1 - y_{vi}) = 1$, as part $u \in V$ belongs to LRU Q_i and part $v \in V$ does not. This corresponds to the definition of $B(Q_i)$. Subsequently, we consider each edge $e \in E$ that needs to be broken before breaking $\{u, v\}$; i.e., for each $\{u, v\} \in B(Q_i)$ we consider all $e \in H(\{u, v\})$. Hence, the variable k_i^e satisfies $k_i^e \geq y_{ui}(1 - y_{vi})$, $\forall \{u, v\} \in E, \forall e \in H(\{u, v\}), \forall i \in \{1, \ldots, |V|\}$. Note that k_i^e may take the value of one, even if an edge $e \in E$ is fully contained within a LRU Q_i. We use the variable k_i^e in our objective function (3.3a), since it represents whether an edge has to be broken ($k_i^e = 1$) in order to remove LRU Q_i. Furthermore, the objective function enforces that $k_i^e = 0$ if edge e is not broken for the replacement of Q_i. Next, we use k_i^e, the edge weights, and the failure rate of Q_i (expressed using y_{vi}) to determine the costs for replacing Q_i. The total purchase costs of the LRU Q_i

are derived by using y_{vi}, and we multiply this by the total failure rate of Q_i. This results in a binary non–linear programming formulation of LRU Design:

$$\text{(BNLP)} \quad \min_{\boldsymbol{Y},\boldsymbol{K}} \sum_{i=1}^{|V|} \sum_{e\in E} k_i^e w(e) \sum_{v\in V} y_{vi}\lambda(v) + \sum_{i=1}^{|V|} \sum_{u\in V} y_{ui}\ell(u) \sum_{v\in V} y_{vi}\lambda(v) \tag{3.3a}$$

$$\text{s.t.} \quad \sum_{i=1}^{|V|} y_{vi} = 1, \qquad \forall v \in V, \tag{3.3b}$$

$$k_i^e \geq y_{ui}(1 - y_{vi}), \qquad \forall \{u,v\} \in E, \forall e \in H(\{u,v\}), \; \forall i \in \{1,\dots,|V|\}, \tag{3.3c}$$

$$y_{vi}, k_i^e \in \{0,1\}. \tag{3.3d}$$

Constraints (3.3b) ensure that each part $v \in V$ is included in exactly one LRU, and constraints (3.3c) enforce the definition of the auxiliary variable k_i^e.

The BNLP is a problem with a quadratic objective function and quadratic constraints (constraints (3.3c)). Therefore, we propose to linearize our problem by applying the McCormick reformulation (McCormick, 1976); i.e., we introduce new variables $\rho_{ev}^i = y_{xi}k_i^e$ and $\sigma_{uv}^i = y_{ui}y_{vi}$. The variable ρ_{ev}^i denotes whether LRU Q_i contains part $v \in V$ and whether edge $e \in E$ needs to be broken in order to replace the LRU Q_i. Analogously, σ_{uv}^i denotes whether two parts $u, v \in V$ are both contained in the same LRU Q_i. Substituting ρ_{ev}^i and σ_{uv}^i into the above implies that we need to add constraints. Furthermore, we optimize over $\boldsymbol{Y}$, $\boldsymbol{K}$, $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$, where $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$ correspond to the 3–D arrays with entries ρ_{ev}^i and σ_{uv}^i, respectively. Hence, we obtain the BLP formulation of LRU Design:

$$\text{(BLP)} \quad \min_{\boldsymbol{Y},\boldsymbol{K},\boldsymbol{\rho},\boldsymbol{\sigma}} \sum_{i=1}^{|V|} \sum_{e\in E} \sum_{v\in V} \rho_{ev}^i \lambda(v) w(e) + \sum_{i=1}^{|V|} \sum_{u,v\in V} \sigma_{uv}^i \ell(u)\lambda(v) \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{i=1}^{|V|} y_{vi} = 1, \qquad \forall v \in V, \tag{3.4b}$$

$$k_i^e \geq y_{ui} - \sigma_{uv}^i, \qquad \forall \{u,v\} \in E, \forall e \in H(\{u,v\}), \; \forall i \in \{1,\dots,|V|\}, \tag{3.4c}$$

$$\rho_{ev}^i \leq y_{vi}, \qquad \forall e \in E, \forall v \in V, \; \forall i \in \{1,\dots,|V|\}, \tag{3.4d}$$

$$\rho_{ev}^i \leq k_i^e, \qquad \forall e \in E, \forall v \in V, \; \forall i \in \{1,\dots,|V|\}, \tag{3.4e}$$

$$\rho_{ev}^i \geq y_{vi} + k_i^e - 1, \qquad \forall e \in E, \forall v \in V, \; \forall i \in \{1,\dots,|V|\}, \tag{3.4f}$$

$$\sigma_{uv}^i \leq y_{ui}, \qquad \forall u, v \in V, \forall i \in \{1,\dots,|V|\}, \tag{3.4g}$$

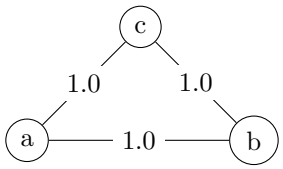
$$\sigma_{uv}^i \leq y_{vi}, \qquad \forall u, v \in V, \forall i \in \{1,\dots,|V|\}, \tag{3.4h}$$

$$\sigma^i_{uv} \geq y_{ui} + y_{vi} - 1, \qquad \forall u, v \in V, \forall i \in \{1, \ldots, |V|\}, \quad (3.4i)$$
$$\rho^i_{ev}, \sigma^i_{uv} \geq 0, \quad (3.4j)$$
$$y_{vi}, k^e_i \in \{0, 1\}. \quad (3.4k)$$

Note that constraints (3.3c) have been altered after substitution. Originally we had $y_{ui}(1 - y_{vi}) = y_{ui} - y_{ui}y_{vi}$, which yields constraints (3.4c) after substitution of $\sigma^i_{uv} = y_{ui}y_{vi}$.

We observe that relaxing the integrality of the decision variable y_{vi} in Problem (3.4) need not yield an optimal integer solution, and thus is infeasible to LRU Design. Example 3.1 illustrates that we may indeed find an optimal fractional solution if we relax integrality for y_{vi} for Problem (3.4).

Example 3.1 Consider the following vertex set $V = \{a, b, c\}$ and edge set $E = \{\{a, b\}, \{b, c\}, \{a, c\}\}$. Furthermore, let $\ell(v) = 1$ for all $v \in V$, $\lambda(a) = \lambda(b) = 2.0$, $\lambda(c) = 2.5$, $w(e) = 1$ for all $e \in E$, and $A = \emptyset$. Graphically, the connection graph is given by:



If we solve this example by Problem (3.4) with relaxed integrality of y_{vi}, we obtain the following optimal solution (columns represent LRUs)

$$\boldsymbol{Y} = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0.5 \end{pmatrix}.$$

◇

3.4. Set partitioning

The BLP formulation of LRU Design may be computationally demanding due to the large number of variables. Therefore, we formulate LRU Design as a set partitioning problem that allows for column generation (branch–and–price) algorithms. A LRU design S consists of various non–intersecting LRUs $Q \in S$ that have been selected. Let $\mathscr{S} = 2^V$ be the power set of V from which LRUs can be selected; $\mathscr{S}$ contains all possible LRUs. Then, $S \subset \mathscr{S}$, and our objective is to determine which solution S is optimal via column generation. A LRU $Q \in \mathscr{S}$ can equivalently be represented as a $(0, 1)$ column with $|V|$ elements, where a 1 indicates that a vertex is in the LRU Q

and a 0 denotes that the vertex does not belong to the LRU Q. Hence, we consider the matrix entries z_{vQ} that equal 1 if $v \in Q$ and 0 otherwise. Then, a column from the matrix $Z = (z_{vQ})$ corresponds to LRU Q, and we denote this column by Z_Q. Note that a column Z_Q and the LRU $Q \subseteq V$ are equivalent representations of a LRU.

Let x_Q be the indicator variable that denotes whether a LRU $Q \in \mathscr{S}$ is selected for the LRU design S. The vector of size $|\mathscr{S}|$ consisting of all x_Q is denoted by x, and based on x we can straightforwardly derive the solution S to LRU Design by $S = \{Q \in \mathscr{S} : x_Q > 0\}$. We remark that S can equivalently be represented as the submatrix $\mathcal{Z} = \{Z_Q : x_Q > 0\}$ of Z. Our objective is to determine the LRU design in terms of x_Q such that the total costs are minimized, and each part $v \in V$ is included in exactly one LRU. We capture this in the Master Problem (M):

$$\text{(M)} \quad \min_{x} \quad \sum_{Q \in \mathscr{S}} \omega(Q) x_Q \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{Q \in \mathscr{S}} z_{vQ} x_Q = 1, \qquad \forall v \in V, \tag{3.5b}$$

$$x_Q \in \{0,1\}, \qquad \forall Q \in \mathscr{S}. \tag{3.5c}$$

The objective function (3.5a) minimizes the total costs of using LRUs, while constraints (3.5b) enforce that each part $v \in V$ is included in exactly one LRU $Q \in \mathscr{S}$. The set $\mathscr{S}$ is exponentially large, so straightforward optimization is not tractable. Therefore, we propose to solve the LP relaxation of M by column generation, and we note that this LP relaxation is a lower bound to M. Thus, we relax integrality of x_Q to obtain the LP relaxation of the Master Problem called LPM. In Section 3.4.1 we prove that there exists an optimal integer solution to LPM. By running the the simplex algorithm, such an optimal integer solution is found and it is thus also optimal for M. Subsequently, we present our procedure for solving LPM in Section 3.4.2.

3.4.1 The relationship between M and LPM

We prove that an optimal integer solution to LPM exists by considering a so–called LRU cycle. We show that if a given fractional solution contains a LRU cycle, there exists a feasible solution to LPM without the LRU cycle and strictly lower costs. This implies that an optimal solution does not contain a LRU cycle. Next, we that there exists an optimal integer solution to LPM and this can be found by running the simplex algorithm.

Let $\tilde{x}$ be a fractional solution to LPM with $\tilde{S} = \{Q \in \mathscr{S} : \tilde{x}_Q > 0\}$ (or equivalently $\tilde{\mathcal{Z}} = \{Z_Q : \tilde{x}_Q > 0\}$) and such that each $Q \in \tilde{S}$ is a connected subgraph of G. Furthermore, let x^* be an optimal solution to LPM with $S^* = \{Q \in \mathscr{S} : x^*_Q > 0\}$ (or equivalently $\mathcal{Z}^* = \{Z_Q : x^*_Q > 0\}$) and that also has connected LRUs $Q \in S^*$. Note that x^* exists by Lemma 3.1.

Definition 3.1 A LRU cycle is a collection of LRUs $C = \{Q_1, Q_2, \ldots, Q_n\}$ such that each Q_i is connected, $n \geq 3$ and for all $1 \leq i \leq n$ we have $Q_i \cap Q_{i+1} \neq \emptyset$, $(Q_i \cap Q_{i+1}) \setminus (Q_{i+1} \cap Q_{i+2}) \neq \emptyset$, $(Q_{i+1} \cap Q_{i+2}) \setminus (Q_i \cap Q_{i+1}) \neq \emptyset$, with $n+1 \equiv 1 \pmod{n}$ and $n+2 \equiv 2 \pmod{n}$.

For an example of a LRU cycle, we refer the reader to Figure 3.3. Next, we use the concept of a LRU cycle to obtain the following result.

Lemma 3.2 *An optimal solution x^* to LPM does not contain a LRU cycle.*

Proof. We show that a solution to LPM that contains a LRU cycle is suboptimal. Let $\tilde{x}$ be a fractional solution to LPM such that each $Q \in \tilde{S}$ is connected and there exists a LRU cycle $C = \{Q_1, Q_2, \ldots, Q_n\} \subseteq \tilde{S}$. Note that a solution that contains a LRU solution must be fractional. We prove that there exists a feasible solution x' to LPM with $S' = \{Q \in \mathscr{S} : x'_Q > 0\}$ in which the LRU cycle C does not exist and $\pi(S') < \pi(\tilde{S})$.

In this proof we focus on LRUs Q_i, Q_{i-1}, and Q_{i+1} with $1 \leq i \leq n$, $n+1 \equiv 1 \pmod{n}$, and $Q_0 \equiv Q_n$. Let us consider what edges are broken when a part $v \in Q_i \cap Q_{i+1}$ fails and we replace LRU $Q_i \in C$. In this case, we break all edges $e \in \Gamma(Q_i)$. Let $\mathcal{F}(X, Y) = \Gamma(X) \setminus \Gamma(Y)$ for sets X, Y. Then, we have $\Gamma(Q_i) = \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1}) \cup (\Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1}))$, since $\mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1}) \subset \Gamma(Q_i)$. Next, we study what happens when a part $v \in Q_i \setminus Q_{i+1}$ fails and we replace Q_i. Analogous to the foregoing, we break all edges $e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1}) \cup (\Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})) = \Gamma(Q_i)$.

Let $W_i = \min\left\{\sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e), \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_i)} w(e)\right\}$ for each LRU Q_i. Furthermore, we define $Q_i = \operatorname{argmin}_{Q_j \in C}\{W_j\}$ and we assume that $W_i = \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e)$ (later we consider $W_i = \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_i)} w(e)$).

We create an alternative solution x' by partitioning Q_i in $Q_i \cap Q_{i+1}$ and $Q_i \setminus Q_{i+1}$. That is, let the alternative solution x' be identical to $\tilde{x}$ except for the entries $x'_{Q_i} = 0$, $x'_{Q_i \setminus Q_{i+1}} = \tilde{x}_{Q_i}$, and $x'_{Q_i \cap Q_{i+1}} = \tilde{x}_{Q_i}$. We have

$$
\begin{aligned}
&\pi(S') - \pi(\tilde{S}) \\
&= \tilde{x}_{Q_i} \Bigg(\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \sum_{u \in Q_i \cap Q_{i+1}} \ell(u) \sum_{v \in Q_i \cap Q_{i+1}} \lambda(v) \\
&\quad + \sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) + \sum_{u \in Q_i \setminus Q_{i+1}} \ell(u) \sum_{v \in Q_i \setminus Q_{i+1}} \lambda(v) \\
&\quad - \sum_{e \in \Gamma(Q_i)} w(e) \sum_{x \in \Gamma(Q_i)} \lambda(x) - \sum_{u \in Q_i} \ell(u) \sum_{v \in Q_i} \lambda(v) \Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&< \tilde{x}_{Q_i} \left(\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \right. \\
&\quad \left. - \sum_{e \in \Gamma(Q_i)} w(e) \sum_{x \in Q_i} \lambda(x) \right) \\
&= \tilde{x}_{Q_i} \left(\left[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i)} w(e) \right] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) \right. \\
&\quad \left. + \left[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i)} w(e) \right] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \right),
\end{aligned}
$$

where the inequality holds as $\sum_{u \in Q_i \cap Q_{i+1}} \ell(u) \sum_{v \in Q_i \cap Q_{i+1}} \lambda(v) + \sum_{u \in Q_i \setminus Q_{i+1}} \ell(u) \sum_{v \in Q_i \setminus Q_{i+1}} \lambda(v) \leq \sum_{u \in Q_i} \ell(u) \sum_{v \in Q_i} \lambda(v)$, because $Q_i \cap Q_{i+1}$ and $Q_i \setminus Q_{i+1}$ partition Q_i and $\lambda(v) > 0, \ell(v) > 0$ for all $v \in V$. We continue by proving that the right hand side of the last equality is less than zero; i.e., we show that $\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) \leq \sum_{e \in \Gamma(Q_i)} w(e)$ and $\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \leq \sum_{e \in \Gamma(Q_i)} w(e)$. Let us rewrite the above as

$$
\begin{aligned}
&\pi(S') - \pi(\tilde{S}) \\
&< \tilde{x}_{Q_i} \left(\left[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i)} w(e) \right] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) \right. \\
&\quad \left. + \left[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i)} w(e) \right] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \right) \\
&= \tilde{x}_{Q_i} \left(\left[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) \right. \right. \\
&\quad \left. - \sum_{e \in \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) \right] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \left[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \right. \\
&\quad \left. \left. - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})} w(e) \right] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \right).
\end{aligned}
$$

The equality holds, because $\mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1}) \subset \Gamma(Q_i)$ and $\mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1}) \subset \Gamma(Q_i)$.

First, we study what happens when we replace $Q_i \cap Q_{i+1}$. We start by proving that $\nexists (e_1, e_2) \in A : e_1 \in \Gamma(Q_{i-1}), e_2 \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})$. Next, we use this in order to show that if $e \in \Gamma(Q_i \cap Q_{i+1})$ then $e \notin \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})$. Finally, we prove that the latter result implies $\Gamma(Q_i \cap Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i) \subseteq \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})$. We use these results to show that $\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) -$

$\sum_{e\in\Gamma(Q_i)\setminus\mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})} w(e) \leq 0$. Next, we do a similar analysis for when we replace $Q_i\setminus Q_{i+1}$ to conclude that $\Gamma(Q_i\setminus Q_{i+1})\setminus\mathcal{F}(Q_i\cap Q_{i+1},Q_i) \subseteq \Gamma(Q_i)\setminus\mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$. We use this result to show that $\sum_{e\in\Gamma(Q_i\setminus Q_{i+1})} w(e) - \sum_{e\in\mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})} w(e) - \sum_{e\in\Gamma(Q_i)\setminus\mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})} w(e) \leq 0$.

Now, consider replacing $Q_i \cap Q_{i+1}$. We have that $\nexists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i-1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$. Suppose, $\exists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i-1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$, then $e_2 \in \Gamma(Q_{i-1})$ as $(e_1,e_2) \in A$. If $e_2 \in \Gamma(Q_i\cap Q_{i-1})$, then $e_2 \notin \Gamma(Q_i\cap Q_{i-1}) \setminus \Gamma(Q_{i-1}) = \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$ as $e_2 \in \Gamma(Q_{i-1})$. Thus, we obtain a a contradiction. On the other hand, if $e_2 \notin \Gamma(Q_i\cap Q_{i-1})$, then $e_2 \notin \Gamma(Q_i\cap Q_{i-1}) \setminus \Gamma(Q_{i-1}) = \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$, which is also a contradiction. Hence, $\nexists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i-1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$.

Next, we show that for each $e_2 \in \Gamma(Q_i\cap Q_{i+1})$, we have $e_2 \notin \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$. Suppose that $e_2 \in \Gamma(Q_i\cap Q_{i+1}) \cap \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$. We assumed that $(\{u,v\},\{v,x\}) \in A$ for $u,v,x \in V$, and therefore we have that each $e_2 = \{u,v\} \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$ satisfies that $u,v \in Q_{i-1}$. Furthermore, if $e_2 \in \Gamma(Q_i\cap Q_{i+1})$ and $e_2 \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$, there exists a path $(e^1,e^2,\ldots,e_2)$ – where each edge of the path belongs to $\Gamma(Q_i\cap Q_{i+1})$ – because $(\{u,v\},\{v,x\}) \in A$ for $u,v,x \in V$ and each $\{u,v\} \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$ satisfies that $u,v \in Q_{i-1}$. But this implies that $\exists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i-1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$, which is a contradiction. Hence, if $e \in \Gamma(Q_i\cap Q_{i+1})$ then $e \notin \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$.

Next, we consider $\Gamma(Q_i\cap Q_{i+1}) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_i)$, and have $\Gamma(Q_i\cap Q_{i+1}) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_i) \subseteq \Gamma(Q_i)$ as $\mathcal{F}(Q_i\cap Q_{i+1},Q_i) = \Gamma(Q_i\cap Q_{i+1}) \setminus \Gamma(Q_i)$. Furthermore, each $e \in \Gamma(Q_i\cap Q_{i+1}) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_i)$ satisfies $e \in \Gamma(Q_i\cap Q_{i+1})$ and thus $e \notin \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$, as we proved in the foregoing. Hence, we have $\Gamma(Q_i\cap Q_{i+1}) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_i) \subseteq \Gamma(Q_i) \setminus \mathcal{F}(Q_i\cap Q_{i-1},Q_{i-1})$.

Next, we consider replacing $Q_i \setminus Q_{i+1}$. By the same reasoning as above we have $\nexists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i+1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$. Now, suppose $e_2 \in \Gamma(Q_i \setminus Q_{i+1}) \cap \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$, then each $e_2 = \{u,v\} \in \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$ satisfies that $u,v \in Q_{i+1}$. Furthermore, $e_2 \in \Gamma(Q_i \setminus Q_{i+1})$ and $e_2 \in \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$ imply that there exists a path $(e^1,e^2,\ldots,e_2)$ – where each edge belongs to $\Gamma(Q_i\setminus Q_{i+1})$ – because $(\{u,v\},\{v,x\}) \in A$ for $u,v,x \in V$ and each $\{u,v\} \in \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$ satisfies that $u,v \in Q_{i+1}$. Hence, $\exists(e_1,e_2) \in A : e_1 \in \Gamma(Q_{i+1}), e_2 \in \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$, which is a contradiction. Hence, if $e \in \Gamma(Q_i \setminus Q_{i+1})$ then $e \notin \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$. Next, we consider $\Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i \setminus Q_{i+1},Q_i) \subseteq \Gamma(Q_i)$, and as each $e \in \Gamma(Q_i \setminus Q_{i+1})$ is such that $e \notin \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$ we have $\Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i \setminus Q_{i+1},Q_i) \subseteq \Gamma(Q_i)\setminus\mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$. Finally, we observe that each $e = \{u,v\} \in \mathcal{F}(Q_i\setminus Q_{i+1},Q_i)$ satisfies $u,v \in Q_i$ and consequently we have that $\mathcal{F}(Q_i\setminus Q_{i+1},Q_i) = \mathcal{F}(Q_i\cap Q_{i+1},Q_i)$ because $Q_i \setminus Q_{i+1}$ and $Q_i \cap Q_{i+1}$ partition Q_i and we break the same edges for both due to the existing symmetry. Hence, we also have $\Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_i) \subseteq \Gamma(Q_i) \setminus \mathcal{F}(Q_i\cap Q_{i+1},Q_{i+1})$. Then, we obtain

$$\pi(S') - \pi(\tilde{S})$$

$$
\begin{aligned}
&< \tilde{x}_{Q_i} \Bigg(\Bigg[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) \\
&\quad - \sum_{e \in \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) \Bigg] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \Bigg[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \\
&\quad - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})} w(e) \Bigg] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \Bigg) \\
&\leq \tilde{x}_{Q_i} \Bigg(\Bigg[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})} w(e) \\
&\quad - \sum_{e \in \Gamma(Q_i \cap Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) \Bigg] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \Bigg[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \\
&\quad - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})} w(e) - \sum_{e \in \Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) \Bigg] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \Bigg) \\
&\leq \tilde{x}_{Q_i} \Bigg(\Bigg[\sum_{e \in \Gamma(Q_i \cap Q_{i+1})} w(e) - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) \\
&\quad - \sum_{e \in \Gamma(Q_i \cap Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) \Bigg] \sum_{x \in Q_i \cap Q_{i+1}} \lambda(x) + \Bigg[\sum_{e \in \Gamma(Q_i \setminus Q_{i+1})} w(e) \\
&\quad - \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) - \sum_{e \in \Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e) \Bigg] \sum_{x \in Q_i \setminus Q_{i+1}} \lambda(x) \Bigg) \\
&= 0.
\end{aligned}
$$

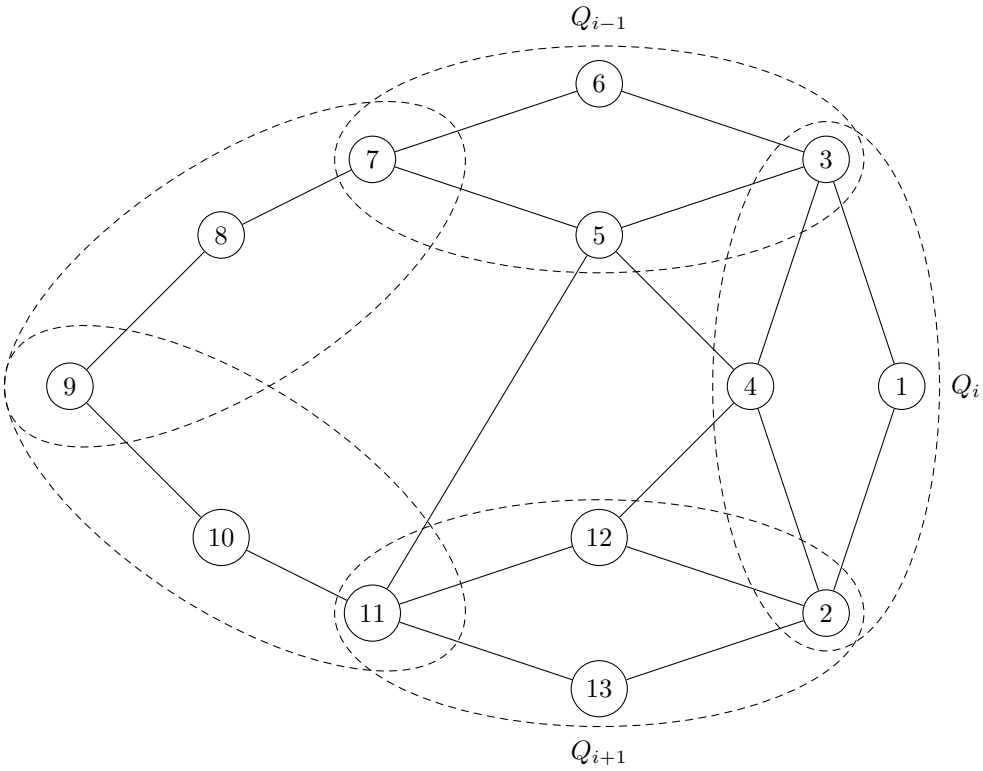
The second inequality holds by the previous observations that $\Gamma(Q_i \cap Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i) \subseteq \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1})$ and $\Gamma(Q_i \setminus Q_{i+1}) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_i) \subseteq \Gamma(Q_i) \setminus \mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1})$. The last inequality follows because $Q_i = \operatorname{argmin}_{Q_j \in C}\{W_j\}$ and we assumed that $W_i = \sum_{e \in \mathcal{F}(Q_i \cap Q_{i+1}, Q_i)} w(e)$. Hence, x' is a solution without the LRU cycle C and satisfies $\pi(S') < \pi(\tilde{S})$.

Next, consider the case $W_i = \sum_{e \in \mathcal{F}(Q_i \cap Q_{i-1}, Q_i)} w(e)$. Then, we create the solution x' by partitioning Q_i in $Q_i \cap Q_{i-1}$ and $Q_i \setminus Q_{i-1}$, and we follow the same procedure as above where we replace Q_{i+1} and Q_{i-1} by Q_{i-1} and Q_{i+1}, respectively. Finally, the solution x' does not have the LRU cycle C and satisfies $\pi(S') < \pi(\tilde{S})$.

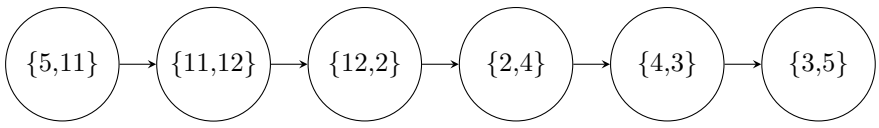
Hence, an optimal solution x^* to LPM does not contain a LRU cycle. □

The result of Lemma 3.2 is illustrated by means of Example 3.2.

Example 3.2 Suppose we have the connection graph G with $w(e) = 1$, $\forall e \in E$ and the precedence graph D from Figure 3.3. Note that the failure rate $\lambda(v)$ and the purchase cost $\ell(v)$ are not relevant for this example. We consider the solution $\tilde{x}$ with a LRU cycle $C \subseteq \tilde{S}$ as drawn by the dashed ellipses in Figure 3.3. We illustrate our procedure for splitting LRU a Q_i into $Q_i \cap Q_{i+1}$ and $Q_i \setminus Q_{i+1}$. In this example, we



(a) Connection Graph



(b) Precedence Graph

Figure 3.3: The input graphs and a LRU cycle C

have

$$\begin{aligned}
\Gamma(Q_i) &= \{\{2,12\},\{2,13\},\{2,4\},\{3,4\},\{3,5\},\{3,6\},\{4,5\},\{4,12\}\},\\
\Gamma(Q_{i+1}) &= \{\{1,2\},\{10,11\},\{4,12\},\{5,11\},\{11,12\},\{12,2\},\{2,4\},\{4,3\},\\
&\quad \{3,5\}\},\\
\Gamma(Q_{i-1}) &= \{\{1,3\},\{4,5\},\{7,8\},\{5,11\},\{11,12\},\{2,12\},\{2,4\},\{3,4\},\\
&\quad \{3,5\}\},\\
\Gamma(Q_i \cap Q_{i-1}) &= \{\{1,3\},\{4,3\},\{3,5\},\{3,6\}\},\\
\Gamma(Q_i \setminus Q_{i-1}) &= \{\{1,3\},\{2,12\},\{2,13\},\{2,4\},\{3,4\},\{3,5\},\{4,5\},\{4,12\}\},\\
\mathcal{F}(Q_i \cap Q_{i+1}, Q_{i+1}) &= \{\{2,12\},\{2,13\}\},\\
\mathcal{F}(Q_i \cap Q_{i-1}, Q_i) &= \{\{1,3\}\},\\
\mathcal{F}(Q_i \cap Q_{i-1}, Q_{i-1}) &= \{\{3,6\}\}.
\end{aligned}$$

One can use the above expressions to verify that the procedure in the proof of Lemma 3.2 yields a solution x' such that $\pi(S') < \pi(\tilde{S})$. ◇

Next, we introduce Lemma 3.3 and Lemma 3.4 that – in combination with Lemma 3.2 – help us to prove there exists an optimal integer solution to LPM. But first, we present the definition of a totally balanced matrix, see for instance Anstee and Farber (1984) and Hoffman et al. (1985).

Definition 3.2 A binary matrix $\mathcal{R}$ is totally balanced if it does not contain a square submatrix R that has no identical columns and the sum of each row and column equals to two.

Lemma 3.3 *Given a binary $k \times k$ matrix R with $k \geq 3$, no identical columns, and such that each row and column sum to two, there exists an $n \times n$ submatrix $\hat{R}$ of R with $n \geq 3$, no identical columns, and such that each row and column of $\hat{R}$ sum to two. If we interpret the columns of $\hat{R}$ as LRUs and the rows of $\hat{R}$ as vertices, then the LRUs – corresponding to the columns of $\hat{R}$ – are a LRU cycle.*

Proof. Consider an $n \times n$ submatrix $\hat{R}$ of R with minimal $n \geq 3$, no identical columns, and such that each row and column of $\hat{R}$ sum to two. Such submatrix $\hat{R}$ exists, because R satisfies the same conditions. We will rename the rows and columns of $\hat{R}$ such that we can easily show that the columns (LRUs) of $\hat{R}$ are a LRU cycle. This renaming procedure is as follows.

The first row is v_1 and Q_1 and Q_2 are the columns such that $\hat{r}_{v_1,Q_1} = \hat{r}_{v_1,Q_2} = 1$. This follows without loss of generality, because $\hat{R}$ is binary and the sum of each row equals two. Furthermore, note that all other values of v_1 are zero. Next, let v_2 be the second row such that $\hat{r}_{v_2,Q_2} = 1$. This is feasible because the sum of column Q_2 is two. Moreover, all other rows (except v_1 and v_2) have the value 0 in column Q_2. We

also remark that $\hat{r}_{v_2,Q_1} = 0$, since otherwise all other values in column Q_1 (except for $\hat{r}_{v_1,Q_1}$ and $\hat{r}_{v_2,Q_1}$) are zero and this means that columns Q_1 and Q_2 are identical, which is a contradiction.

Next, we label the column Q_i such that $\hat{r}_{v_{i-1},Q_i} = 1$ for each $i = 3, \ldots, n$, and we call the row v_i that satisfies $\hat{r}_{v_i,Q_i} = 1$, for all $i = 3, \ldots, n$. This can be done due to the following reasoning. The columns Q_j with $1 < j < i - 1$ are such that $\hat{r}_{v_{i-1},Q_j} = 0$, because each column Q_j already sums to two. Unless $i = n$, we have $\hat{r}_{v_{i-1},Q_1} = 0$ because otherwise we would have a $i \times i$ submatrix for which each row and column sum equal 2, $i \geq 3$, and where no identical columns exists. But this would contradict the fact that n is minimal. Hence, we can label Q_i such that $\hat{r}_{v_{i-1},Q_i} = 1$. Moreover, all rows v_j with $1 \leq j < i - 1$ are such that $\hat{r}_{v_j,Q_i} = 0$, because each row v_j already sums to two (by considering columns Q_k with $k < i$). Therefore, we can call a row v_i such that $\hat{r}_{v_i,Q_i} = 1$.

Finally, we let $r_{v_n,Q_1} = 1$ such that the row and column sum of *each* row and column of $\hat{R}$ equal 2.

Given the renaming of the columns and rows of $\hat{R}$, we have for all $1 \leq i \leq n$ that $\{v_i\} = Q_i \cap Q_{i+1}$ with $n + 1 \equiv 1 \pmod{n}$. Furthermore, this implies that $(Q_i \cap Q_{i+1}) \setminus (Q_{i+1} \cap Q_{i+2}) = \{v_i\} \setminus \{v_{i+1}\} \neq \emptyset$ and $(Q_{i+1} \cap Q_{i+2}) \setminus (Q_i \cap Q_{i+1}) = \{v_{i+1}\} \setminus \{v_i\} \neq \emptyset$ with $n + 1 \equiv 1 \pmod{n}$ and $n + 2 \equiv 2 \pmod{n}$. But this implies that the LRUs $Q_1, Q_2, \ldots, Q_n$ are a LRU cycle. □

The result of Lemma 3.3 is directly related to the definition of a totally balanced matrix, see Definition 3.2.

Lemma 3.4 *If an optimal solution x^* does not contain a LRU cycle, then $\mathcal{Z}^*$ is totally balanced.*

Proof. Given an optimal solution x^* – with S^* or equivalently $\mathcal{Z}^*$ – that does not contain a LRU cycle, there does not exist a binary $k \times k$ submatrix R of $\mathcal{Z}^*$ with $k \geq 3$, no identical columns, and such that the sum of each row and column of R equals to two. This follows by the contraposition of Lemma 3.3. Hence, the matrix $\mathcal{Z}^*$ is totally balanced. □

Finally, we use Lemma 3.3 and Lemma 3.4 to establish Theorem 3.1.

Theorem 3.1 *There exists an optimal integer solution to LPM.*

Proof. Let x^* be an optimal solution to LPM. Each LRU $Q \in S^*$ is a connected subgraph of G by Lemma 3.1 and x^* does not contain a LRU cycle by Lemma 3.2. Then, the matrix $\mathcal{Z}^*$ is totally balanced by Lemma 3.4. Consequently, the polyhedron $\mathcal{P} = \left\{x : \mathcal{Z}^* x = 1, x \geq 0, x \in \mathbb{R}^{|\mathcal{S}^*|}\right\}$ is integral (Fulkerson and Hoffman,

1974). Hence, x^* is either integer or a convex combination of integer solutions to LPM, and thus we obtain our desired result. □

If LPM is solved by the simplex algorithm, we obtain an optimal solution x^* that is an extreme point of the polyhedron $P = \left\{x : Zx = 1, x \geq 0, x \in \mathbb{R}^{|\mathscr{S}|}\right\}$, but is also an extreme point of the polyhedron $\mathcal{P} = \left\{x : \mathcal{Z}^* x = 1, x \geq 0, x \in \mathbb{R}^{|\mathcal{S}^*|}\right\}$ spanned by the submatrix $\mathcal{Z}^*$. Theorem 3.1 now implies that x^* is integral, because x^* is an extreme point of $\mathcal{P}$. Hence, solving LPM with the simplex algorithm yields an optimal integer solution, and this solution is thus also optimal for M.

We would like to stress the fact that the polyhedron P of LPM is *not* integral, but we still obtain an optimal integer solution. There exist solutions with a LRU cycle that are an extreme point of P, since we consider the power set $\mathscr{S} = 2^{|V|}$. Hence, P is not integral. We are able to obtain an optimal integer solution to LPM, because we prove suboptimality of the extreme points (solutions) that contain a LRU cycle, see Lemma 3.2. Such an approach contrasts with much other research that focuses on proving integrality of a polyhedron to conclude that an optimal integer solution can be found (if the objective function is convex). We demonstrate that – for non–integral polyhedra – analysis of the objective function can be used to establish the existence of an optimal integer solution to a relaxed problem when the full constraint matrix will not guarantee the existence of an optimal integer solution.

The result of Theorem 3.1 can be generalized, see Corollary 3.1.

Corollary 3.1 *If the objective function of LPM is convex and LRU cycles are suboptimal, then there exists an optimal integer solution to LPM.*

Proof. Let x^* be an optimal solution to LPM, where LPM has a convex objective function and is such that LRU cycles are suboptimal. Then, x^* does not contain a LRU cycle, and thus $\mathcal{Z}^*$ is totally balanced by Lemma 3.4. Hence, the polyhedron $\mathcal{P} = \left\{x : \mathcal{Z}^* x = 1, x \geq 0, x \in \mathbb{R}^{|\mathcal{S}^*|}\right\}$ is integral, and thus x^* is either integer or a convex combination of integer solutions to LPM. Therefore, we obtain our desired result. □

Corollary 3.1 implies – similar to Theorem 3.1 – that solving LPM by the simplex algorithm yields an optimal integer solution that is also optimal for M (given that M has the same convex objective function). This results from the convexity of the objective function.

3.4.2 Solving LPM and M

Given Theorem 3.1, we move our attention to solving LPM, for which we apply column generation. Hence, we consider a feasible subset of LRUs (or columns) $\tilde{\mathscr{S}} \subseteq \mathscr{S}$ for

LPM. This results in the *Restricted Master Program* (RLPM). For RLPM, we generate profitable LRUs (columns) by solving the pricing problem of RLPM:

$$c^* = \min_{Q \in \mathscr{S}} \left\{ \omega(Q) - \sum_{v \in Q} r_v \right\}, \tag{3.6}$$

where r_v are dual variables for the partitioning constraints of RLPM. We want to find a LRU $Q \in \mathscr{S}$ with minimal reduced costs. We can find this LRU by using a problem formulation similar to the BNLP from Problem (3.3), where we also consider an auxiliary variable k^e representing whether we disconnect edge $e \in E$ for the LRU. Note that $k^e = 0$ for an edge that is *not* disconnected, by the objective function (3.7a). We introduce the binary decision variable γ_v that determines whether a part $v \in V$ is included in the LRU. Hence, we rewrite Problem (3.6) to obtain

$$\begin{aligned}
c^* = \quad & \min_{\boldsymbol{\gamma}, \boldsymbol{k}} && \sum_{e \in E} k^e w(e) \sum_{v \in V} \gamma_v \lambda(v) + \sum_{u \in V} \gamma_u \ell(u) \sum_{v \in V} \gamma_v \lambda(v) - \sum_{v \in V} \gamma_v r_v && && \text{(3.7a)} \\
& \text{s.t.} && k^e \geq \gamma_u (1 - \gamma_v), && \forall \{u, v\} \in E, \forall e \in H(\{u, v\}), && \text{(3.7b)} \\
& && \gamma_v, k^e \in \{0, 1\}. && && \text{(3.7c)}
\end{aligned}$$

Linearizing Problem (3.7) by substituting $\eta_{ev} = k^e \gamma_v$ and $\delta_{uv} = \gamma_u \gamma_v$ yields the final form of the pricing problem (3.8). We use the same McCormick linearization method from Section 3.3. Note that η_{ev} denotes whether the LRU contains part $v \in V$ and that edge $e \in E$ needs to be broken for the LRU to be removed. Similarly, δ_{uv} represents whether the LRU contains both parts $u, v \in V$.

$$\begin{aligned}
c^* = \quad & \min_{\boldsymbol{k}, \boldsymbol{\gamma}, \boldsymbol{\eta}, \boldsymbol{\delta}} && \sum_{e \in E} \sum_{v \in V} \eta_{ev} \lambda(v) w(e) + \sum_{u, v \in V} \delta_{uv} \ell(u) \lambda(v) - \sum_{v \in V} \gamma_v r_v && && \text{(3.8a)} \\
& \text{s.t.} && k^e \geq \gamma_u - \delta_{uv}, && \forall \{u, v\} \in E, \forall e \in H(\{u, v\}), && \text{(3.8b)} \\
& && \eta_{ev} \leq k^e, && \forall e \in E, \forall v \in V, && \text{(3.8c)} \\
& && \eta_{ev} \leq \gamma_v, && \forall e \in E, \forall v \in V, && \text{(3.8d)} \\
& && \eta_{ev} \geq k^e + \gamma_v - 1, && \forall e \in E, \forall v \in V, && \text{(3.8e)} \\
& && \delta_{uv} \leq \gamma_u, && \forall u, v \in V, && \text{(3.8f)} \\
& && \delta_{uv} \leq \gamma_v, && \forall u, v \in V, && \text{(3.8g)} \\
& && \delta_{uv} \geq \gamma_u + \gamma_v - 1, && \forall u, v \in V, && \text{(3.8h)} \\
& && \eta_{ev}, \delta_{uv} \geq 0, \gamma_v, k^e \in \{0, 1\}. && && \text{(3.8i)}
\end{aligned}$$

After we solve the pricing problem (3.8), we add the obtained LRU to $\tilde{\mathscr{S}}$ and we again solve LPM with the new $\tilde{\mathscr{S}}$. Next, we solve the pricing problem again, and we repeat this procedure until the pricing problem does not return a profitable LRU (column), i.e., we terminate when $c^* \geq 0$. This means that there does not exist a LRU (column) that is worthwhile to add to our LPM, and we have obtained the optimal solution.

3.5. Numerical experiments

In this section, we use the binary linear program formulation and the set partitioning formulation of LRU Design to gain some insight in the size of instances that can be solved and we explore the effects of parameter perturbations on our model's outcomes. We have implemented all optimization model formulations in JuMP (Lubin and Dunning, 2015; Dunning et al., 2017), which is a mathematical optimization package of Julia (Balbaert et al., 2016), and we solved all problems by using Gurobi 7.0.1 on an Intel i5–4300U @2.50GHz processor with 16GB RAM and running Ubuntu 16.04 LTS.

In Section 3.5.1, we explain how we generate our instances for the experiment, and in Section 3.5.2 we study the difference between the computation times (in seconds) of the binary linear programming formulation and the set partitioning formulation. Furthermore, we shed some light on the effect that the downtime costs per time unit have on the LRU design: how many LRUs are used in an optimal solution as the downtime costs increase. Moreover, we study how the system's complexity affects the total annual costs by considering the number of connections between parts, and the number of predecessor–successor relationships that exist.

3.5.1 Instance generator

An instance is described by the graphs G and D. We vary the number of vertices $|V|$, the number of edges $|E|$, and the number of arcs $|A|$ in our numerical experiments. We relate the number of edges in G to the number of vertices by $|E| = \delta|V|$, where δ is the average vertex degree in the graph G. Similarly, we relate the number of arcs in D to the number of edges by $|A| = \delta_E|E|$, where δ_E is the average out degree of an edge $e \in E$. All other parameters such as $\lambda(v) > 0$ and $\ell(v) > 0$ for all $v \in V$, and $w(e) > 0$ for all $e \in E$ are randomly generated, as well as a graph's layout in terms of the edge set E and the arc set A.

The graphs G and D are generated in the following way. For G, we have a set of vertices V and a number of unique edges $|E| = \delta|V|$, and we create a spanning tree with $|V| - 1$ edges. We add an arbitrary vertex $v \in V$ to a set of considered vertices $\tilde{V}$, and we select a new vertex $u \in V \setminus \tilde{V}$ and connect it to an arbitrary vertex $z \in \tilde{V}$ by adding the edge $\{u, v\}$ to the edge set E. We keep doing this until $\tilde{V} = V$. Subsequently we add remaining edges randomly to our graph and we terminate once we have $|E|$ edges in G. Secondly, we generate the precedence graph D. We (randomly) assign an index to each edge $e \in E$ and denote this index by $I(e)$, and the minimum and maximum values assigned are 1 and $|E|$, respectively. We start with $A = \emptyset$ and add an arc in each iteration. An iteration starts by selecting two random edges $\{u, v\}, \{v, x\} \in E : u, v, x \in V$ and $u \neq v \neq x$. If $I(\{u, v\}) \leq I(\{v, x\})$ we create an arc $(\{u, v\}, \{v, x\})$ and add it to A, otherwise we create an arc $(\{v, x\}, \{u, v\})$ and

add it to A. We repeat this procedure until $\delta_E|E| = |A|$, and upon termination we have obtained a set A that has a topological sorting and thus the precedence graph D is acyclic.

3.5.2 Computational results

Next, we discuss the computational results for our model. The generation of a random graph follows the procedure from Section 3.5.1, and we let $|V| \in \{10, 20, 30, 40, 50, 60\}$, $\delta \in \{2, 3, 4\}$, and $\delta_E \in \{0.5, 1, 1.5\}$. For each combination $(|V|, \delta, \delta_E)$, we generate 10 random instances, resulting in a total of 540 instances.

We use a time limit of 600 seconds for the BLP formulation and also for the set partitioning formulation. This time limit is relatively low because we solve a large number of instances, thereby making it feasible to perform the entire numerical study in a reasonable amount of time. If an instance has not been solved to optimality within 600 seconds, we say that it is inefficient. If all instances of a certain parameter combination $(|V|, \delta, \delta_E)$ are inefficient, we write – as an entry for the combination. We determine the average computation time of both formulations based on the efficient instances. The results are presented in Table 3.2, where the computation times are given in seconds, and the subscripts indicate the number of efficient instances. Furthermore, we have not reported computation times for the BLP with $|V| \geq 40$, since we have found no efficient solutions within the time limit.

We observe that, given the time limit of 600 seconds, the BLP formulation can only solve small size instances up to 20 vertices (parts), while the set partitioning (SP) formulation can solve medium size to large instances up to 60 vertices (parts). Furthermore, we see that the set partitioning formulation solves instances faster than the BLP formulation. This effect is amplified when the instances become harder, i.e., when $|V|$, δ and δ_E increase. Real–life instances are typically medium to large sized instances and can have 50 vertices (parts). Furthermore, such instances may possess many and complex connections and predecessor–successor relationships. This makes the BLP formulation unsuitable for practical purposes. Hence, it is worthwhile to invest extra time to implement the set partitioning formulation (LPM) with a pure pricing algorithm. Furthermore, the computation times illustrate that the set partitioning formulation of LRU Design is particularly useful as a feedback mechanism for the OEM's design department. The engineers can quickly assess many design alternatives (in terms of the connection graph and precedence graph) and their effects on the optimal LRU design and the corresponding (after–sales) costs.

Next, we numerically study the effect of the cost of one time unit of system downtime on the number of LRUs that is used in an optimal LRU design. In the remainder of this section, we use the same instance generator as discussed in Section 3.5.1, and we generate 1,000 instances per parameter setting (δ, δ_E) and keep $|V| = 20$. For a given instance, we vary the edge weights by multiplying all edge weights of the instance by a constant factor $q \in \{0.1, 1, 10\}$. A higher value for q means that it is more expensive

BLP		$\lvert V\rvert$		
δ	δ_E	10	20	30
2	0.5	4.67_{10}	159.90_{8}	–
2	1	5.92_{10}	170.76_{5}	–
2	1.5	3.89_{10}	124.68_{8}	–
3	0.5	8.69_{10}	353.53_{6}	–
3	1	7.80_{10}	–	–
3	1.5	5.77_{10}	230.76_{3}	–
4	0.5	10.28_{10}	570.27_{1}	–
4	1	9.31_{10}	–	–
4	1.5	8.10_{10}	249.06_{1}	–

SP		$\lvert V\rvert$					
δ	δ_E	10	20	30	40	50	60
2	0.5	0.16_{10}	1.07_{10}	8.63_{10}	18.93_{10}	143.65_{10}	342.21_{9}
2	1	0.50_{10}	2.64_{10}	18.88_{10}	51.33_{10}	330.65_{9}	491.22_{5}
2	1.5	0.16_{10}	0.85_{10}	5.61_{10}	16.22_{10}	86.17_{10}	252.03_{10}
3	0.5	0.29_{10}	1.82_{10}	12.07_{10}	18.52_{10}	171.50_{10}	359.44_{5}
3	1	0.79_{10}	5.63_{10}	16.92_{10}	75.69_{10}	266.78_{4}	447.94_{2}
3	1.5	0.44_{10}	1.82_{10}	8.36_{10}	23.89_{10}	235.61_{10}	528.73_{5}
4	0.5	0.37_{10}	2.51_{10}	7.83_{10}	29.83_{10}	332.28_{8}	469.10_{1}
4	1	0.93_{10}	6.30_{10}	12.22_{10}	99.56_{10}	497.46_{3}	–
4	1.5	0.77_{10}	3.40_{10}	14.13_{10}	28.89_{10}	450.83_{6}	–

Table 3.2: Average computation times (sec) of both formulations

to break edges. If the time for breaking an edge remains constant, it means that the cost rate per time unit for breaking an edge increases, and thus we can capture a higher downtime cost per time unit by varying q. This way, we create three classes of instances (i) low downtime cost per time unit ($q = 0.1$); (ii) moderate downtime cost per time unit ($q = 1$); (iii) and high downtime cost per time unit ($q = 10$). We keep the parameter values for δ and δ_E constant at $\delta = 3$ and $\delta_E = 1$. We focus on the number of LRUs $|S^*|$ in an optimal solution S^*. The results are presented in Figure 3.4.

Based on Figure 3.4, the instances where the downtime cost per time unit is low, have many small LRUs (each part is a LRU in itself in the extreme case). These solutions prefer small LRUs because they have lower purchase costs. As the cost for a single time unit of downtime increases, we see that the optimal solution prefers fewer LRUs that each become larger, because such larger LRUs enable faster replacement and thus lower downtime costs. This explains, for example, why we observe that the consumer electronics industry with low values for q has rather small LRUs. On the other end of the spectrum, capital intensive industries such as the semiconductor industry or the

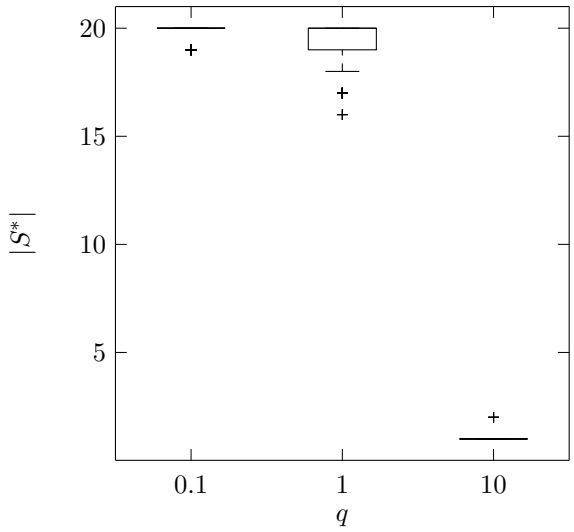


Figure 3.4: Effect of edge weights on the number of LRUs in S^*

aviation industry have high values for q, and they tend to opt for larger LRUs which enable faster replacement. Both these phenomena are confirmed by the numerical results of our model.

The second effect that we study considers the complexity of the system, and how this affects the costs of the optimal LRU design. We restrict our attention, for now, on the number of edges in the connection graph G that describes system complexity. We vary how strongly various parts are connected to each other by altering δ. A low (high) value of δ corresponds to lesser (more) connected parts. We are interested in the effect that the number of connections in G has on the costs, because this provides a justification of whether to avoid many connections between parts in order to reduce the total costs. For our analysis, we keep δ_E and q constant at $\delta_E = 1$ and $q = 1$. The results are presented in Figure 3.5.

We observe that more connections in the connection graph G result in cost increases, because we need to disconnect more edges in order to remove a LRU. This has an important managerial implication, as engineers should be urged to reduce the number of connections in systems to be developed. Thus, it may be wise for an OEM to invest extra in a system's design such that the number of connections in G is reduced. An example wherein few number of connections lead to low costs is a bicycle. A typical connection graph of bicycles has few connections, and consequently relatively low replacement costs because we only need to disconnect few connections in case a part fails.

Finally, we also study the effect that system complexity has on the costs of the optimal

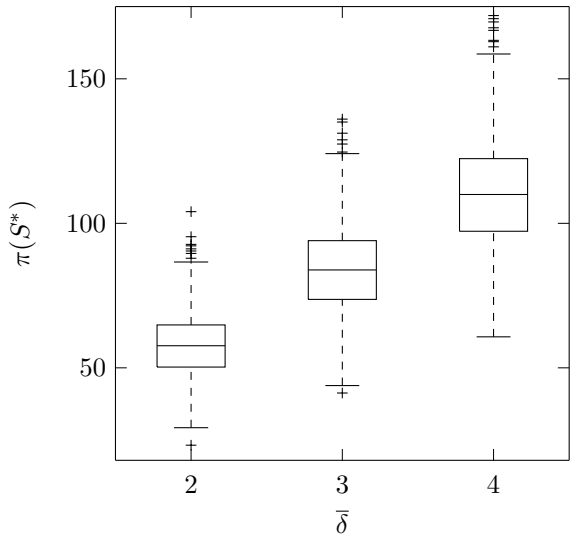


Figure 3.5: Effect of the number of connections in G on $\pi(S^*)$

solution $\pi(S^*)$, when we consider the number of predecessor–successor relationships. A lower value of δ_E indicates that fewer predecessor–successor relationships exist in the precedence graph D. Similar to the foregoing, we keep the other parameters constant at $\delta = 3$ and $q = 1$. The results for different values of δ_E are depicted in Figure 3.6 and we observe a similar behavior to changes in δ.

The costs increase as the number of predecessor–successor relationships increases, because we need to disconnect more connections upon the failure of a LRU. Consequently, the costs of an optimal solution $\pi(S^*)$ increase when the number of connections in D increases (as δ_E increases). The managerial implications of our results also align with those for δ: managers should urge their designers to avoid predecessor–successor relationships in order to reduce costs. This objective may be easier to attain than avoiding connections in the connection graph G by careful design. Hence, the results confirm that careful design (in terms of G and D) is crucial to reduce the overall costs.

3.6. Conclusions

We considered an OEM that is concerned with the maintenance of a system. If the system does not operate, the OEM loses money, customer goodwill or has to pay customers a downtime penalty. Therefore, the OEM is interested in lowering the cost for non–functioning systems by designing Line Replaceable Units (LRUs) that can

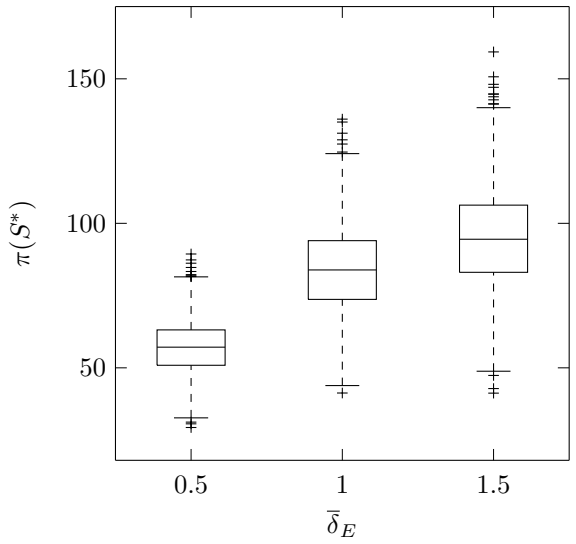


Figure 3.6: Effect of the number of predecessor–successor relationships in D on $\pi(S^*)$

be removed quickly. Furthermore, the LRUs should not be too large, because this increases a LRU's total failure rate and the LRU's purchase costs (or repair costs). Thus, the OEM has to determine what the optimal LRU design is that balances the replacement costs and the purchase costs (or repair costs) of LRUs.

We presented a novel approach for representing the connections between parts in a system, also capturing the existing disassembly sequences. This system representation can be used as a visual aid to enhance internal communication at the OEM, between the design department and operations department. In this chapter, we used the system representation to derive an optimization model that minimizes the replacement costs and the purchase costs (or repair costs) by optimizing the LRU design. We called this problem LRU Design, and we formulated it as a binary linear program and as a set partitioning problem. The latter formulation allows for branch–and–price algorithms. We proved that the set partitioning formulation can be solved to optimality by a pure pricing algorithm and this results in an optimal integer solution, i.e., branching is unnecessary. This reduces the computation times and thus makes the set partitioning formulation of LRU Design particularly useful as a feedback mechanism for the OEM's design department. The engineers can quickly assess various design alternatives (in terms of the connection graph and precedence graph) and their effects on the optimal LRU design and the corresponding (after–sales) costs. Furthermore, the set partitioning formulation is suitable to solve large instances, while the binary linear programming formulation is not. In addition to the computation times, we observed that optimal solutions to LRU Design have larger LRUs when the cost per time unit of system downtime increases, because this

enables faster replacement and thus avoids large downtime costs. Finally, we found that managers should urge their designers to reduce the number of connections and predecessor–successor relationships in a system's design.

3.A. Deriving $H(e)$

We determine $H(e)$ for all edges $e \in E$ in polynomial time by the following polynomial algorithms, where Algorithm 2 is called by Algorithm 1.

Algorithm 1 Derive $H(e)$ for all edges $e \in E$

```
1: procedure RemovalEdges(E, A)
2:     Ê ← E
3:     Ẽ ← Degree(D(Ê, A))
4:     while Ẽ ≠ ∅ do
5:         for all e ∈ Ẽ do
6:             H(e) ← {e}
7:         end for
8:         for all e ∈ Ẽ do
9:             for all (e, z) ∈ A do
10:                H(e) ← H(e) ∪ H(z)
11:            end for
12:        end for
13:        Ê ← Ê \ Ẽ
14:        Ẽ ← Degree(D(Ê, A))
15:    end while
16:    return H(e) for all e ∈ E
17: end procedure
```

Algorithm 2 Determine all edges $e \in E$ that have no successors in D

```
1: procedure Degree(D)
2:     Ẽ ← ∅
3:     for all e ∈ E do
4:         if δ^out(e) = 0 then
5:             Ẽ ← Ẽ ∪ {e}
6:         end if
7:     end for
8:     return Ẽ
9: end procedure
```

4

Design of non–disjoint Line Replaceable Units

4.1. Introduction

This chapter is an extension to Chapter 3. The motivation of using LRUs and the related literature are therefore identical to Chapter 3, and we do not discuss these here. Furthermore, our system representation is identical to Chapter 3 and consists of a connection graph and a precedence graph. However, we do relax the assumption that each part belongs to exactly one Line Replaceable Unit (LRU). Although this assumption has practical advantages, e.g. it may reduce the number of possible LRUs and therefore can yield lower complexity for organizational processes such as inventory planning, we see that parts oftentimes belong to more than one LRU in practice.

For example, if we consider the laptop example from Chapter 3, we may want to replace the motherboard and graphics card together when the motherboard fails, but we only want to replace the graphics card in case the graphics card fails. For the failure of both parts (motherboard and graphics card), the graphics card is replaced and thus the graphics card belongs to more than one LRU. Another example one can think of is a bicycle. Consider the rear wheel assembly that consists of a tire, rim, spokes, wheel hub, and the cassette. Suppose that we replace a spoke by a new one if the spoke fails (a spoke is a LRU). Furthermore, we replace the entire rear wheel (the tire, rim, all spokes, wheel hub, and the cassette) if the rim breaks. Then, a spoke belongs to two LRUs.

By means of the previous two examples, we observe that there may exist parts in

systems that belong to more than one LRU. The benefit of having parts that belong to more than one LRU is that we may be able to further reduce the total costs. This is a result from the extra freedom that we obtain by relaxing the assumption that each part belongs to exactly one LRU. Therefore, we consider a LRU problem wherein parts can belong to more than one LRU in this chapter. We call this problem C–LRU Design, where the addition C stands for cover, because our model is a cover model rather than a partition model from Chapter 3.

We prove that C–LRU Design is separable in the parts of the system. In addition, we numerically find that the computation times of C–LRU Design are low, even for large instances. The computation times of the set partitioning formulation of LRU Design can be more than 45 times higher than the computation times (on average) of C–LRU Design. This is a direct result of the separable nature of C–LRU Design. Furthermore, we find that C–LRU Design has the potential to reduce costs for systems with few parts and a high number of connections and predecessor–successor relationships. Examples of such systems may be laptops (Lenovo) or bicycles (Gazelle). For more complex systems such as lithography systems (ASML), trains (NedTrain), or trucks (PACCAR/Volkswagen), the cost benefit of C–LRU Design is limited. Finally, we find that C–LRU Design can use more or fewer LRUs than LRU Design.

This chapter is organized similar to Chapter 3. In Section 4.2, we present our model where parts can belong to more than one LRU. We also show each LRU is connected in any optimal solution to C–LRU Design. Furthermore, we show that C–LRU Design is decomposable in the parts of the system, and we use this decomposition result to formulate C–LRU Design as multiple binary non–linear programs (v–BNLP) in Section 4.3. Moreover, we propose a linearization for each of the binary non–linear programs to obtain a binary linear programming (BLP) formulation. We do not discuss a column generation approach because this is unnecessary due to the separability result. In Section 4.4, we numerically study the computation times of the BLP formulation of C–LRU Design. We also study the effects of parameter perturbations on the results of C–LRU Design. Finally, we compare LRU Design to C–LRU Design in terms of computation times, costs, and number of LRUs used in an optimal solution. We conclude this chapter in Section 4.5.

4.2. Model

We represent a system by the connection graph $G = (V, E)$ and the precedence graph $D = (E, A)$, with V, E and A corresponding to the vertex set, edge set and arc set, respectively. Furthermore, we let $\lambda(v) > 0$ and $\ell(v) > 0$ be the failure rate and purchase cost of part $v \in V$, and we define $w(e) > 0$ as the cost of breaking edge $e \in E$. An arc $(\{u, v\}, \{v, x\}) \in A$ denotes that we have to break $\{v, x\}$ prior to breaking $\{u, v\}$.

A part $v \in V$ belongs to at least one LRU. That is, part v is replaced upon its own failure, but it may also be replaced upon the failure of a part $u \in V : u \neq v$. Therefore, we represent a LRU differently from Chapter 3. We let a LRU be a tuple characterized by a replacement set and a failure set, i.e., $Q = (R_Q, F_Q)$ where Q is the LRU, $R_Q \subseteq V$ is the replacement set, and $F_Q \subseteq R_Q$ is the failure set. The failure of a part in the failure set triggers replacement of the LRU. The replacement set is replaced if any of the vertices in the failure set fails. For example, if we consider the laptop example with the motherboard and the graphics card, there exists a LRU with a replacement set consisting of the graphics card and the motherboard together, while the failure set consists of the motherboard (if the motherboard fails we replace the motherboard and graphics card together). Furthermore, we assume that a part $v \in V$ belongs to exactly one failure set, i.e., the failure sets partition V.

Next, we study what happens when a LRU Q fails, or technically what happens when a part $v \in F_Q$ fails. In this case, we have to break all edges $e \in \Gamma(R_Q)$. We determine $\Gamma(R_Q)$ similar to Chapter 3 by $\Gamma(R_Q) = \bigcup_{e \in B(R_Q)} H(e)$, and using Algorithms 1 and 2 from Chapter 3.

Now, we turn our attention to the failure rate and purchase costs of a LRU Q. Each LRU Q has a failure rate of $\sum_{v \in F_Q} \lambda(v)$, because all parts $v \in F_Q$ induce the replacement of R_Q. Similarly, the total purchase cost of LRU Q is given by $\sum_{v \in R_Q} \ell(v)$.

Analogous to the derivation of $\omega(Q)$ in Chapter 3, the total costs of a LRU Q is given by

$$\omega_c(Q) = \sum_{e \in \Gamma(R_Q)} w(e) \sum_{v \in F_Q} \lambda(v) + \sum_{u \in R_Q} \ell(u) \sum_{v \in F_Q} \lambda(v).$$

We are interested in determining the optimal LRU design. Let S_c be a collection of LRUs such that $\emptyset \notin S_c$ and each part $v \in V$ is included in at least one replacement set and in exactly one failure set; i.e., $\bigcup_{Q \in S_c} F_Q = V$, $F_Q \cap F_{Q'} = \emptyset$ for all $Q, Q' \in S_c : F_Q \neq F_{Q'}$, and $F_Q \subseteq R_Q$ for each LRU $Q \in S_c$. We would like to underline that we use different notation for a LRU design in this chapter, i.e., S_c instead of S, because a LRU Q is a tuple in this chapter. The total costs of a LRU design S_c are then given by

$$\pi_c(S_c) = \sum_{Q \in S_c} \omega_c(Q) = \sum_{Q \in S_c} \sum_{b \in \Gamma(R_Q)} w(b) \sum_{v \in F_Q} \lambda(v) + \sum_{Q \in S_c} \sum_{u \in R_Q} \ell(u) \sum_{v \in F_Q} \lambda(v). \tag{4.1}$$

Next, we define C–LRU DESIGN as: What is the LRU Design S_c that minimizes $\pi_c(S_c)$?

We can generalize the model for C–LRU DESIGN such that a subset of parts are correctively maintained, while others are preventively maintained. This generalization is analogous to Remark 3.1. Similarly, it is possible to extend C–LRU DESIGN such that failed parts are repaired, but this is again analogous to Remark 3.2.

Next, we explore structure in the optimal solution S_c^* in Proposition 4.1, which is a similar result to Lemma 3.1.

Proposition 4.1 *Each replacement set R_Q of LRU $Q = (R_Q, F_Q) \in S_c^*$ is connected subgraph of G, for any optimal solution S_c^* to* C–LRU Design.

Proof. Let $\mathscr{J}_R$ be the finite set of connected components in the subgraph of G induced by the replacement set R_Q of a LRU Q, and $|\mathscr{J}_R| \geq 1$. If $|\mathscr{J}_R| = 1$ the replacement set R_Q is connected, which satisfies our claim. Hence, we consider $|\mathscr{J}_R| \geq 2$ in the remainder. Then, $\Gamma(\mathcal{J}_R) \subset \Gamma(R_Q)$ for any $\mathcal{J}_R \in \mathscr{J}_R$, and we have $\Gamma(R_Q) = \bigcup_{\mathcal{J}_R \in \mathscr{J}_R} \Gamma(\mathcal{J}_R)$ as $\bigcup_{\mathcal{J}_R \in \mathscr{J}_R} \mathcal{J} = R_Q$. Next, let $\mathscr{J}_F$ be a set of failure sets; we construct a failure set $\mathcal{J}_F \subseteq \mathcal{J}_R$ for each $\mathcal{J}_R \in \mathscr{J}_R$, and all failure sets $\mathcal{J}_F$ partition F_Q. Then, $\sum_{\mathcal{J}_F \in \mathscr{J}_F} \sum_{v \in \mathcal{J}_F} \lambda(v) = \sum_{v \in R_Q} \lambda(v)$, because of the partition. So, each replacement set $\mathcal{J}_R \in \mathscr{J}_R$ is associated to a failure set $\mathcal{J}_F \in \mathscr{J}_F$. This corresponds to a LRU $\mathcal{J} = (\mathcal{J}_R, \mathcal{J}_F)$, and we let $\mathscr{J}$ be the set of all LRUs $\mathcal{J}$. Furthermore, we note that $\mathcal{J}_R = R_\mathcal{J}$ and $\mathcal{J}_F = F_\mathcal{J}$ by definition of a LRU $\mathcal{J}$. Then,

$$\begin{aligned}
\sum_{\mathcal{J} \in \mathscr{J}} \omega_c(\mathcal{J}) &= \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(R_\mathcal{J})} w(e) \sum_{v \in F_\mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in R_\mathcal{J}} \ell(u) \sum_{v \in F_\mathcal{J}} \lambda(v) \\
&\leq \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(R_Q)} w(e) \sum_{v \in F_\mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in R_\mathcal{J}} \ell(u) \sum_{v \in F_\mathcal{J}} \lambda(v) \\
&< \sum_{\mathcal{J} \in \mathscr{J}} \sum_{e \in \Gamma(R_Q)} w(e) \sum_{v \in F_\mathcal{J}} \lambda(v) + \sum_{\mathcal{J} \in \mathscr{J}} \sum_{u \in R_Q} \ell(u) \sum_{v \in F_\mathcal{J}} \lambda(v) \\
&= \sum_{e \in \Gamma(R_Q)} w(e) \sum_{v \in F_Q} \lambda(v) + \sum_{u \in R_Q} \ell(u) \sum_{v \in F_Q} \lambda(v) = \omega_c(Q),
\end{aligned}$$

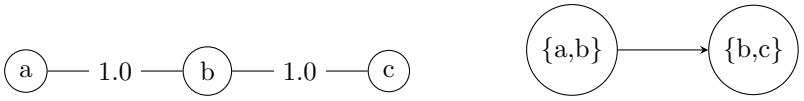
where the first inequality follows from the fact that each $\Gamma(\mathcal{J}_R) = \Gamma(R_\mathcal{J}) \subset \Gamma(R_Q)$, $\forall \mathcal{J} \in \mathscr{J}$. The second inequality follows from the fact that $R_\mathcal{J} \subset R_Q$, $\forall \mathcal{J} \in \mathscr{J}$, and thus $\sum_{u \in R_\mathcal{J}} \ell(u) < \sum_{u \in R_Q} \ell(u)$, $\forall \mathcal{J} \in \mathscr{J}$. The final equality holds, since the failure sets $F_\mathcal{J}$ partition the vertices in F_Q. Hence, the replacement set R_Q of each LRU $Q \in S_c^*$ is a connected subgraph of G, for any optimal solution S_c^*. □

Although an optimal solution has connected replacement sets, we do not know whether the optimal solution will assign parts to multiple replacement sets. In fact, we show in Example 4.1 that this can occur in an optimal solution to C–LRU Design.

Example 4.1 Let $V = \{a, b, c\}$ with values $\lambda(a) = \lambda(b) = \lambda(c) = 1$, $\ell(a) = 1.2$, $\ell(b) = 0.5$, $\ell(c) = 1.0$. The connection graph and the precedence graph are depicted in Figure 4.1, and we let $w(\{a, b\}) = w(\{b, c\}) = 1.0$.

There exist four different LRU designs such that each $v \in V$ is included in exactly one failure set and in exactly one replacement set:

$S_1 = \{(\{a, b, c\}, \{a, b, c\})\}$; $\pi_c(S_1) = 8.1$
$S_2 = \{(\{a, b\}, \{a, b\}); (\{c\}, \{c\})\}$; $\pi_c(S_2) = 7.4$



(a) Connection Graph *(b)* Precedence Graph

Figure 4.1: Input Graphs

$S_3 = \{(\{a\},\{a\});(\{b,c\},\{b,c\})\}$; $\pi_c(S_3) = 10.6$
$S_4 = \{(\{a\},\{a\});(\{b\},\{b\});(\{c\},\{c\})\}$; $\pi_c(S_1) = 7.7$.

However, if we consider the LRU design $S_5 = \{(\{a,b\},\{a\});(\{b\},\{b\});(\{c\},\{c\})\}$ in which part b belongs to two replacement sets, but to one failure set, we obtain

$$\begin{aligned}\pi_c(S_5) =& w(\{b,c\})\lambda(a) + [w(\{a,b\}) + w(\{b,c\})]\,\lambda(b) + w(\{b,c\})\lambda(c)\\ &+ [\ell(a) + \ell(b)]\,\lambda(a) + \ell(b)\lambda(b) + \ell(c)\lambda(c) \qquad = 7.2\end{aligned}$$

Hence, we see that an optimal solution to this example assigns at least one part to more than one replacement set. ◇

Finally, we make an important observation that C–LRU Design is separable in the parts $v \in V$. We let $R(v)$ be a replacement set for a vertex v such that for a LRU $Q \in S_c$ we have $R_Q = R(v)$, $\forall v \in F_Q$; then we have the following.

Theorem 4.1 C–LRU Design *is equivalent to determining the optimal replacement set* $R(v)$ *for each* $v \in V$.

Proof. For each solution S_c, a LRU $Q = (R_Q, F_Q) \in S_c$ satisfies $R_Q = R(v)$, $\forall v \in F_Q$ by definition of $R(v)$. Hence, we write

$$\begin{aligned}\pi_c(S_c) &= \sum_{Q\in S_c}\sum_{e\in\Gamma(R_Q)} w(e)\sum_{v\in F_Q}\lambda(v) + \sum_{Q\in S_c}\sum_{u\in R_Q}\ell(u)\sum_{v\in F_Q}\lambda(v)\\ &= \sum_{Q\in S_c}\sum_{v\in F_Q}\lambda(v)\left[\sum_{e\in\Gamma(R_Q)} w(e) + \sum_{u\in R_Q}\ell(u)\right]\\ &= \sum_{Q\in S_c}\sum_{v\in F_Q}\lambda(v)\left[\sum_{e\in\Gamma(R(v))} w(e) + \sum_{u\in R(v)}\ell(u)\right]\\ &= \sum_{v\in V}\lambda(v)\left[\sum_{e\in\Gamma(R(v))} w(e) + \sum_{u\in R(v)}\ell(u)\right],\end{aligned}$$

where the second equality follows after rearranging terms. The third equality holds as $R_Q = R(v)$ for $Q \in S_c$ and $\forall v \in F_Q$. The final equality follows because all F_Q

partition V. Hence, our problem is separable in $v \in V$. □

4.3. Binary programming formulation

We use Theorem 4.1 to solve C–LRU Design as $|V|$ binary non–linear programs (BNLP). For each $v \in V$ we present a binary non–linear program that determines the optimal replacement set for this v. Subsequently, we linearize each of these BNLPs. If we solve all $|V|$ binary linear programs (BLPs), we obtain the optimal solution to C–LRU Design. Next, we explain our BNLP and BLP formulations for a given vertex $v \in V$. We determine the optimal replacement set $R^*(v)$ and we introduce the binary variable y_u^v that denotes whether part $u \in V$ belongs to replacement set $R(v)$ for $v \in V$. Hence,

$$y_u^v = \begin{cases} 1 & \text{if } u \in R(v) \\ 0 & \text{otherwise} \end{cases}, \quad \forall u \in V,$$

We refer to $\boldsymbol{y}^v$ as the vector with entries y_u^v. We also define the auxiliary variable $k^e \in \{0,1\}$, which takes the value 1 if edge $e \in E$ is broken for the replacement of $R(v)$. We determine k^e by using the same logic as discussed in Section 3.3. Therefore, we consider the constraint $k^e \geq y_u^v(1 - y_x^v),\ \forall\{u,x\} \in E,\ \forall e \in H(\{u,x\})$ and we add k^e to the objective function. As a consequence, k^e is zero if edge e is *not* broken for the replacement of $R(v)$. Furthermore, $R(v)$ must contain v and this implies that we consider the constraint $y_v^v \geq 1$. Using y_u^v, k^e, and the constraints accordingly we obtain the following binary non–linear program that determines the optimal replacement set $R^*(v)$ for a given $v \in V$:

$$\begin{aligned}
(v\text{–BNLP}) \qquad \min_{\boldsymbol{y}^v, \boldsymbol{k}} \quad & \sum_{e \in E} k^e w(e) \lambda(v) + \sum_{u \in V} y_u^v \ell(u) \lambda(v) && && \text{(4.2a)} \\
\text{s.t.} \quad & k^e \geq y_u^v (1 - y_x^v), && \forall \{u,x\} \in E, \forall e \in H(\{u,x\}) && \text{(4.2b)} \\
& y_v^v \geq 1, \\
& y_u^v, k^e \in \{0,1\}.
\end{aligned}$$

Although v–BNLP has a linear objective function (4.2a), the constraints (4.2b) are quadratic. Hence, we linearize problem by applying the McCormick reformulation (McCormick, 1976). Let $\tau_{ux}^v = y_u^v y_x^v$, which denotes whether both parts $u, x \in V$ are contained in $R(v)$. Now, we optimize over $\boldsymbol{y}^v$, $\boldsymbol{k}$ and the matrix $\boldsymbol{\tau}^v$; and we obtain the binary linear program v–BLP.

$$(v\text{–BLP}) \qquad \min_{\boldsymbol{y}^v, \boldsymbol{k}, \boldsymbol{\tau}^v} \quad \sum_{e \in E} k^e w(e) \lambda(v) + \sum_{u \in V} y_u^v \ell(u) \lambda(v)$$

$$
\begin{aligned}
\text{s.t.} \quad & k^e \geq y_u^v - \tau_{ux}^v, && \forall \{u,x\} \in E, \forall e \in H(\{u,x\}) \\
& y_v^v \geq 1, \\
& \tau_{ux}^v \leq y_u^v, && \forall u,x \in V, \\
& \tau_{ux}^v \leq y_x^v, && \forall u,x \in V, \\
& \tau_{ux}^v \geq y_u^v + y_x^v - 1, && \forall u,x \in V, \\
& \tau_{ux}^v \geq 0, \\
& y_u^v, k^e \in \{0,1\}.
\end{aligned}
$$

For each $v \in V$ we solve v–BLP which results in the optimal replacement sets $R^*(v)$. From all $R^*(v)$ we are able to derive the optimal solution S_c^* to C–LRU Design in a straightforward manner.

We do not present a set covering formulation that allows for column generation algorithms, because we are able to decompose C–LRU Design into $|V|$ subproblems (v–BLP). Each of the subproblems v–BLP is analogous to a pricing problem of the set covering formulation. Hence, there is little added value to consider such a formulation further.

4.4. Numerical experiments

In this section, we numerically study the binary linear programming formulation of C–LRU Design. Analogous to Chapter 3, the BLPs are implemented in JuMP (Lubin and Dunning, 2015; Dunning et al., 2017) – a mathematical optimization package of Julia (Balbaert et al., 2016) – and we solved the models by using Gurobi 7.0.1 on an Intel i5–4300U @2.50GHz processor with 16GB RAM and running Ubuntu 16.04 LTS.

The instances used for our numerical experiments are the same as the instances used in Section 3.5.2. Hence, we only discuss the computational results in the remainder. We start in Section 4.4.1 by studying the computation times of C–LRU Design, the effect of the cost for one time unit of downtime on the optimal LRU design, and the effect of a system's complexity on the total annual costs. Additionally, we compare LRU Design to C–LRU Design with respect to the computation times, the costs, and the number of LRUs used in an optimal solution, in Section 4.4.2.

4.4.1 Results of C–LRU Design

First, we study the computation times and we use the 540 instances from Section 3.5.2 with a time limit of 600 seconds. All instances are solved to optimality within 600 seconds. The presented computation times are the average over the 10 instances for a given parameter combination $(|V|, \delta, \delta_E)$.

		$\|V\|$					
δ	δ_E	10	20	30	40	50	60
2	0.5	0.03	0.16	0.53	1.34	2.57	4.68
2	1	0.04	0.26	0.66	1.54	2.82	5.01
2	1.5	0.04	0.32	0.94	1.86	3.55	6.15
3	0.5	0.04	0.18	0.58	1.43	2.71	4.88
3	1	0.06	0.33	0.80	1.86	3.23	5.69
3	1.5	0.06	0.41	1.29	2.91	4.56	7.99
4	0.5	0.04	0.20	0.64	1.54	2.81	5.00
4	1	0.06	0.35	1.06	2.16	3.59	6.19
4	1.5	0.08	0.53	1.72	5.34	7.12	10.15

Table 4.1: Average computation times (sec) of C–LRU Design

The results in Table 4.1 indicate that the C–LRU Design can be solved efficiently by writing the problem as $|V|$ binary linear programs. Even for large instances ($|V| = 60$, $\delta = 4$, $\delta_E = 1.5$) the computation times are low. However, the computation times increase as the instances become harder ($|V|$, δ, or δ_E increase).

Next, we analyze how the per time unit downtime cost affects the number of LRUs $|S_c^*|$ used in an optimal solution S_c^*. The downtime costs are those costs that are incurred because a system cannot operate. In the remainder of this Section 4.4.1, we use the 1,000 instances that we generated in Chapter 3 per parameter setting (δ, δ_E) and keep $|V| = 20$. We fix $\delta = 3$ and $\delta_E = 1$, and we vary the edge weights of an instance by multiplying the edge weights by a constant $q \in \{0.1, 1, 10\}$. A higher value for q means that it is more expensive to break edges. Given that the time for breaking an edge is constant, this implies that the cost rate per time unit for breaking an edge increases. Hence, we can capture a higher downtime cost per time unit by increasing q. A higher value for q thus corresponds to a higher downtime cost per time unit.

The results from Figure 4.2 are in line with those observed for LRU Design (see Figure 3.4). High downtime costs per time unit (applicable to industries using capital intensive systems) – corresponding to a high value of q – imply that the optimal solution is more geared towards larger LRUs that enable faster replacement rather than smaller LRUs that are cheaper from a purchasing perspective. On the other hand, low cost per time unit of downtime (typically present in the consumer electronics industry) results in smaller LRUs that reduce the purchase costs.

Subsequently, we study the effect that a system's complexity has on the costs of an optimal solution $\pi_c(S_c^*)$. First, we vary the number of connections in the connection graph G and we obtain the results from Figure 4.3. A larger number of connections increases the total costs, because more connections are disconnected (on average) when replacing a LRU. These results are in line with the findings from LRU Design

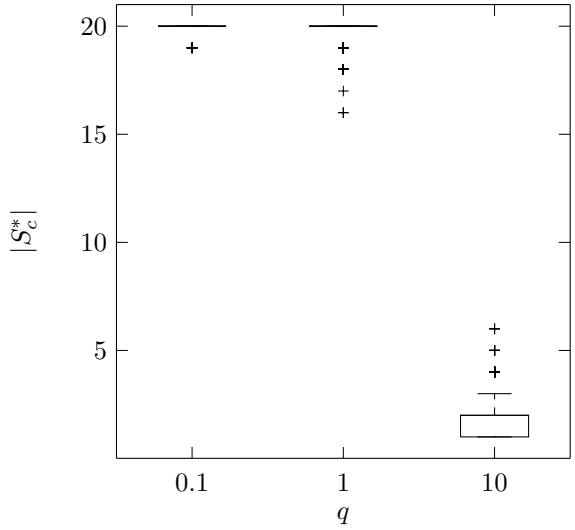


Figure 4.2: Effect of edge weights on the number of LRUs in S_c^*

(see Figure 3.5).

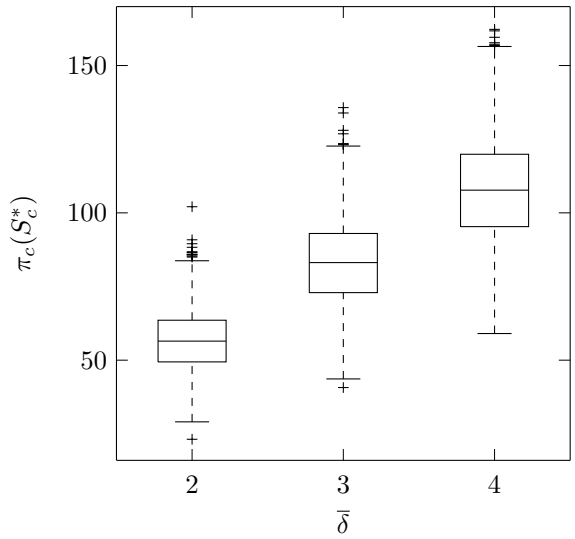


Figure 4.3: Effect of the number of connections in G on $\pi_c(S_c^*)$

Finally, we also address the impact of increasing the number of predecessor–successor relationships, i.e., increasing the number of arcs in the precedence graph D. The results are given in Figure 4.4. An increased number of predecessor–

successor relationships increases the costs, because more more connections have to be disconnected (on average) for the replacement of a LRU. These findings are consistent with the results that we obtained in Figure 3.6 in Chapter 3.

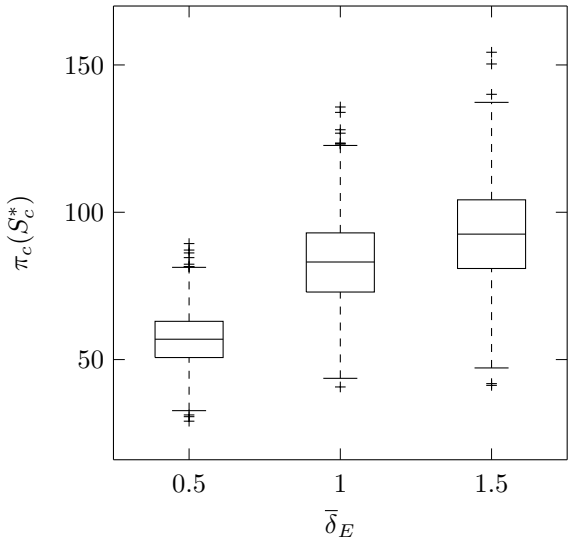


Figure 4.4: Effect of the number of predecessor–successor relationships in D on $\pi_c(S_c^*)$

As our conclusions are consistent with the results from Chapter 3, we also obtain the same managerial implication: the manager is incentivized to urge his designers to reduce the number of connections and the number of predecessor–successor relationships.

4.4.2 Comparing LRU Design to C–LRU Design

Next, we are interested in comparing LRU Design to C–LRU Design with respect to the computation times, cost differences, and the difference in the number of LRUs used in an optimal solution. We compare the set partitioning formulation of LRU Design to the $|V|$ binary linear programs of C–LRU Design.

We use a subset of the 540 instances from Section 3.5.2. We limit our analysis to parameter combinations $(|V|, \delta, \delta_E)$ such that $|V| \leq 40$. For these parameter combinations, the column generation algorithm, which is a pure pricing algorithm, (for LRU Design) is efficient for all instances: all instances of LRU Design are solved to optimality within 600 seconds. We compare C–LRU Design to LRU Design per parameter combination $(|V|, \delta, \delta_E)$ on three dimensions: the relative increase in computation time if we use LRU Design instead of C–LRU Design; the relative cost increase if we use LRU Design instead of C–LRU Design; and

the relative increase in the number of LRUs used in an optimal solution to LRU Design compared to C–LRU Design. Let t and t_c be the average computation time (in seconds) of a parameter combination $(|V|, \delta, \delta_E)$ for LRU Design and for C–LRU Design, respectively. The relative increase in the computation time for such a parameter combination is then given by $\Delta_t = \frac{t-t_c}{t_c} \times 100\%$. The relative increase in the costs for a parameter combination is $\Delta_\pi = \frac{\pi(S^*)-\pi_c(S_c^*)}{\pi_c(S_c^*)} \times 100\%$, where S^* and S_c^* are the optimal solutions of LRU Design and C–LRU Design, respectively. Finally, the relative increase in the number of used LRUs is given by $\Delta_S = \frac{|S^*|-|S_c^*|}{|S_c^*|} \times 100\%$.

The results for the relative increase in the computation times, costs and the number of used LRUs are given in Tables 4.2–4.5. We study average values, minimums, and maximums, which are abbreviated by avg, min, and max, respectively. The average computation times of LRU Design increase compared to C–LRU Design, and this increase can be as high as 4555% ($>$ 45 times the computation time of C–LRU Design). These large increases are a result from partitioning behavior of LRU Design and from the lack of separability in LRU Design. This makes the optimization of the design of LRUs much more difficult and thus more time consuming. The minimum and maximum computation time increase that we observe are 51.56% and 13,017.54%.

		Δ_t	Δ_π		Δ_S		
δ	δ_E	avg (%)	avg (%)	max (%)	avg (%)	min (%)	max (%)
2	0.5	375.83	1.18	4.48	1.50	-10.00	25.00
2	1	1126.38	3.72	10.52	-3.86	-22.22	12.50
2	1.5	264.58	3.75	11.68	15.28	0.00	66.67
3	0.5	645.28	1.64	6.54	-7.67	-66.67	0.00
3	1	1304.28	6.22	16.28	-1.17	-85.71	150.00
3	1.5	622.13	8.98	26.67	11.67	-66.67	100.00
4	0.5	779.73	2.11	9.72	-13.33	-83.33	25.00
4	1	1451.52	4.40	11.47	-19.94	-83.33	66.67
4	1.5	895.37	10.42	18.53	-9.93	-83.33	100.00
All		829.46	4.71	26.67	-3.05	-85.71	150.00

Table 4.2: Differences between LRU Design and C–LRU Design for $|V| = 10$

Second, we study the difference in the costs between C–LRU Design and LRU Design. LRU Design yields strictly higher costs, because this model has less optimization freedom than C–LRU Design. Furthermore, the LRUs of LRU Design are also considered in C–LRU Design. The cost increase may be substantial if we use LRU Design instead of C–LRU Design (up to 26.67%), particularly when the number of parts $|V|$ is low. If there exist fewer parts, the total costs are lower and therefore the extra freedom that C–LRU Design has over LRU Design results in a cost increase when using LRU Design. So for systems with few components (such

		Δ_t	Δ_π		Δ_S		
δ	δ_E	avg (%)	avg (%)	max (%)	avg (%)	min (%)	max (%)
2	0.5	582.20	0.74	1.56	-1.00	-5.00	0.00
2	1	958.14	1.62	4.36	-4.00	-15.00	0.00
2	1.5	162.83	0.26	1.05	0.00	0.00	0.00
3	0.5	930.98	0.56	1.36	-1.00	-5.00	0.00
3	1	1513.57	0.84	3.53	-3.00	-10.00	0.00
3	1.5	341.04	2.37	13.56	2.82	0.00	17.00
4	0.5	1157.53	0.00	0.00	0.00	0.00	0.00
4	1	1721.86	1.58	6.07	0.29	-5.00	17.65
4	1.5	536.87	4.22	13.48	9.30	0.00	25.00
All		878.34	1.34	13.56	0.38	-15.00	25.00

Table 4.3: Differences between LRU Design and C–LRU Design for $|V| = 20$

		Δ_t	Δ_π		Δ_S		
δ	δ_E	avg (%)	avg (%)	max (%)	avg (%)	min (%)	max (%)
2	0.5	1568.33	1.07	3.63	-3.33	-6.67	0.00
2	1	2746.71	2.23	5.49	-5.02	-10.00	0.00
2	1.5	498.19	0.19	0.62	-0.33	-3.33	0.00
3	0.5	1960.26	0.63	1.73	-2.00	-10.00	0.00
3	1	2013.69	0.87	2.45	-2.67	-10.00	0.00
3	1.5	552.40	0.56	2.56	0.71	0.00	7.14
4	0.5	1127.46	0.13	1.20	0.00	0.00	0.00
4	1	1049.63	1.02	3.78	-0.33	-3.33	0.00
4	1.5	708.39	1.80	6.36	3.66	0.00	15.38
All		1358.34	0.94	6.36	-1.04	-10.00	15.38

Table 4.4: Differences between LRU Design and C–LRU Design for $|V| = 30$

as a laptop), LRU Design can substantially increase the costs compared to C–LRU Design. This effect is additional to the increased computation times of LRU Design. Furthermore, there can still exist a notable cost increase for LRU Design when a larger number of parts is considered, e.g. we observe a maximum cost increase of 6.36% and 4.20% for $|V| = 30$ and $|V| = 40$, respectively.

Finally, we consider the difference in the number of LRUs that are used in an optimal design for LRU Design and C–LRU Design. One may expect that C–LRU Design uses strictly more LRUs in an optimal solution, because it has more freedom in selecting the best design of LRUs, i.e., the solution space C–LRU Design is a superset of the solution space of LRU Design. However, the numerical results above show that the number of LRUs used in an optimal solution for C–LRU Design can be

		Δ_t	Δ_π		Δ_S		
δ	δ_E	avg (%)	avg (%)	max (%)	avg (%)	min (%)	max (%)
2	0.5	1320.53	1.20	4.00	-2.53	-10.00	0.00
2	1	3357.23	2.52	4.20	-5.27	-10.00	0.00
2	1.5	764.64	0.19	0.51	-0.25	-2.50	0.00
3	0.5	1203.42	0.54	1.11	-0.75	-5.00	0.00
3	1	4067.25	1.48	3.12	-2.25	-5.00	0.00
3	1.5	737.41	0.36	2.34	0.26	0.00	2.56
4	0.5	1876.65	0.08	0.26	-0.25	-2.50	0.00
4	1	4555.21	0.47	1.39	-1.00	-5.00	0.00
4	1.5	649.42	1.32	3.08	1.82	0.00	5.26
All		2059.08	0.91	4.20	-1.14	-10.00	5.26

Table 4.5: Differences between LRU Design and C–LRU Design for $|V| = 40$

both higher and lower than LRU Design. This is a consequence of the partitioning behavior that is embedded in LRU Design. We illustrate this by two examples in Appendix 4.A, where we compare the number of used LRUs for both models. Lastly, we observe that the difference in the number of used LRUs for C–LRU Design and LRU Design can be large, particularly when the number of parts is low. For higher number of parts, the difference between the used LRUs is smaller, but this may still be as high as 10%.

Overall, C–LRU Design has the desirable properties that the computation times and costs are lower compared to LRU Design. This means that C–LRU Design is preferred if the number of LRUs do not play a large role for decision making. If this number of LRUs does play a large role, we suggest managers to consider both C–LRU Design and LRU Design.

4.5. Conclusion

In this chapter, we extended the model from Chapter 3 such that parts can belong to more than one Line Replaceable Unit (LRU). We let a LRU be a tuple that consists of a failure set and a replacement set. Using this tuple representation, we followed a similar approach to Chapter 3.

We formulated C–LRU Design that minimizes the replacement and purchase (or repair) costs by optimizing the design of LRUs. We proved that C–LRU Design is separable in the parts of the system. Therefore, we determined the optimal replacement set for each individual part. The natural formulation for each these problems is a binary non–linear program, which we linearized to obtain a binary linear program. In our numerical experiments, we illustrated that our formulation

of C–LRU Design – consisting of multiple binary linear programs – can be solved efficiently and thus is suitable for practical instances. Furthermore, we saw that C–LRU Design behaves the same under parameter perturbations as LRU Design (from Chapter 3): optimal solutions have more LRUs when the cost per time unit of system downtime increases, and managers should incentivize their designers to reduce the number of connections and predecessor–successor relationships. Finally, we compared C–LRU Design and LRU Design with respect to the computation times, costs, and the number of LRUs that are used in an optimal solution. We concluded that LRU Design yields significantly higher computation times than C–LRU Design, which can be 4555% higher. Moreover, we observed that LRU Design yields strictly higher costs than C–LRU Design and this increase can be as large as 26.67%. This cost difference, however, decreases to approximately 1% when a system consists of more parts. Furthermore, we saw that C–LRU Design may differ from LRU Design with respect to the number of LRUs that are designed; and this difference can be notable. C–LRU Design does not necessarily increase or reduce the number of LRUs used compared to LRU Design. So, if managers are indifferent on the number of LRUs that are used, then C–LRU Design has clear benefits over LRU Design. On the other hand, if the number of used LRUs matters, we recommend to explore both models in order to enhance decision making.

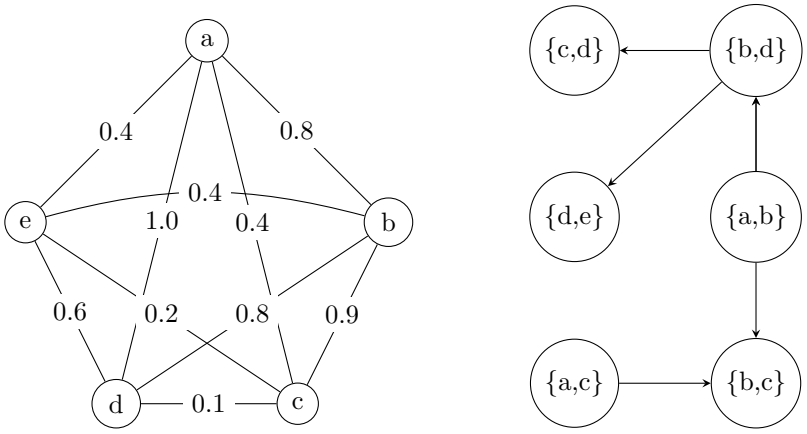
4.A. Numerical examples

In this appendix, we consider Example 4.2 and Example 4.3. For each example, we compare the optimal solutions of LRU Design and C–LRU Design. Example 4.2 is such that the number of LRUs used in an optimal solution is lower under C–LRU Design, while Example 4.3 is such that this number of used LRUs in an optimal solution is higher for C–LRU Design.

Example 4.2 Let $V = \{a, b, c, d, e\}$, and the failure rate and purchase costs for each part are given in Table 4.6. Furthermore, the connection graph with the edge weights $w(e)$ and the precedence graph are given in Figure 4.5.

	a	b	c	d	e
$\lambda(v)$	0.3	0.9	0.3	0.5	0.7
$\ell(v)$	0.8	1.6	0.3	0.5	1.2

Table 4.6: Values for $\lambda(v)$ and $\ell(v)$ for each $v \in V$



(a) Connection Graph *(b)* Precedence Graph

Figure 4.5: Input Graphs

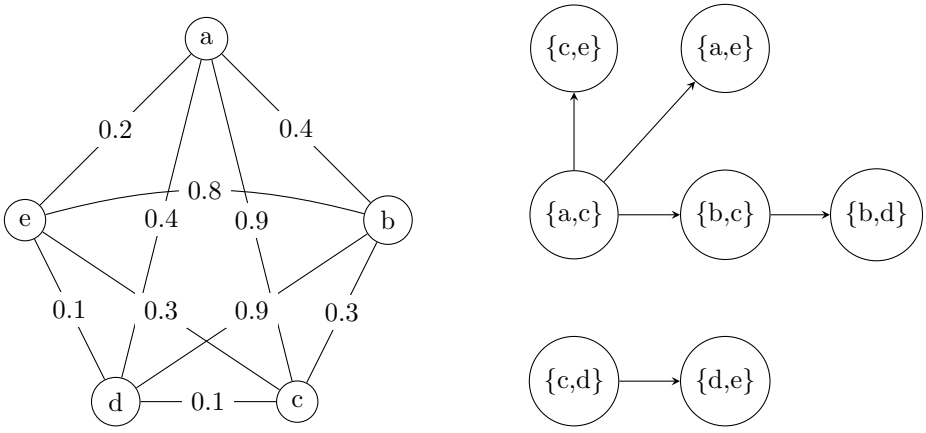
If we solve the above instance for LRU Design, we obtain the optimal solution $S^* = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$. On the other hand, solving the above instance by for C–LRU Design yields the optimal solution $S_c^* = \{(\{a, b, c, d, e\}, \{a, b\}), (\{c\}, \{c\}), (\{d\}, \{d\}), (\{e\}, \{e\})\}$. If a or b fails most edges have to be broken which is expensive, while few edges have to be broken when c, d, or e fails. Therefore, S_c^* replaces all parts if a or b fails because it is cheaper than disconnecting a or b from the other parts, and it

replaces parts c, d, and e individually since these can be replaced cheaply. S^* does not have this freedom, and replaces each part individually because the purchase costs of parts are relatively high. As a result, fewer LRUs are used in an optimal solution to C–LRU Design, i.e., $|S_c^*| = 4 < 5 = |S^*|$. ◇

Example 4.3 Let $V = \{a, b, c, d, e\}$, and the failure rate and purchase costs for each part are given in Table 4.7. Furthermore, the connection graph with the edge weights $w(e)$ and the precedence graph are given in Figure 4.6.

	a	b	c	d	e
$\lambda(v)$	0.4	0.5	0.8	0.4	0.4
$\ell(v)$	0.8	0.1	0.6	0.3	1.1

Table 4.7: Values for $\lambda(v)$ and $\ell(v)$ for each $v \in V$



(a) Connection Graph *(b)* Precedence Graph

Figure 4.6: Input Graphs

If we solve the above instance for LRU Design, we obtain the optimal solution $S^* = \{\{a, b, c, d, e\}\}$. On the other hand, solving the above instance by for C–LRU Design yields the optimal solution $S_c^* = \{(\{a, b, c, d, e\}, \{a, c\}), (\{b\}, \{b\}), (\{d\}, \{d\}), (\{e\}, \{e\})\}$. The purchase costs of the parts are relatively low. Therefore, S^* combines all parts in a single LRU. The optimal solution to C–LRU Design S_c^* can exploit the fact that when a or c fails, most edges are broken. Hence, we observe that we replace all parts if a or c fails. Moreover, S_c^* replaces b, d, and e individually because few edges have to be disconnected for these parts. As a consequence, more LRUs are used in an optimal solution to C–LRU Design, i.e., $|S_c^*| = 4 > 1 = |S^*|$. ◇

5

Implementation of system modifications

5.1. Introduction

In this chapter we take the perspective of an original equipment manufacturer (OEM) that has closed a service contract, e.g. a performance based contract (Cohen et al., 2006), with its customers. As a consequence, the OEM is rewarded (penalized) for better (worse) performance. Examples of OEMs that close such service contracts are encountered in capital goods industries, and include for instance Océ (industrial printing industry) and Pratt & Whitney (aviation industry). In the former example, the OEM (Océ) earns revenue per printed page and by improving the printing speed the OEM is rewarded for better performance. In the latter example the OEM (Pratt & Whitney) sells jet engines and he is paid per flying hour. If he improves the reliability of the engines, the OEM is rewarded for better performance, because the engines operate longer without failures.

The systems of the OEM that are produced and installed are used for a remaining finite lifetime, in the order of magnitude of 10–30 years. In addition, the OEM has closed a service contract with the customers of the several systems for the same remaining finite lifetime. We assume that the closed contracts are identical for all customers. Each system consists of multiple parts. In this chapter, we study multiple critical and repairable parts, and each part occurs once in a system, but this can easily be generalized. If a part fails, the system fails as a result, and therefore the OEM keeps some parts on stock in order to respond quickly to failures. The parts that are currently installed and on stock are called *old parts* (for reasons that will become

clear later). At a certain moment, the OEM believes that the performance of the old parts is insufficient, i.e., they may generate insufficient revenue or they fail too often. Therefore, the OEM can decide to redesign the old component, resulting in a new component with better performance in terms of the failure rate and/or the revenue rate (and it has the same form, fit, function). Returning to our printing example of Océ, the OEM can develop a component that increases the production speed and thus results in more revenue per time unit, because more pages can be printed per time unit and the OEM is rewarded per printed page. Similarly, if the failure rate is reduced, more pages are printed over some finite time period and thus Océ earns more profit. Considering the example of Pratt & Whitney, if the failure rate is reduced, the engine is more in operation and thus generates more profit for the OEM.

As a consequence of the potential performance improvement, the OEM has to determine whether it is profitable to replace the old parts (installed and on stock) by new ones, and if it is he has to decide when to replace the old parts (installed and on stock) by ones. Furthermore, he has to consider the fact that the systems will be used for a finite remaining lifetime. Thus, the new parts are obsolete once the systems reach their end of their life and they will have to be salvaged (for instance sold on a second hand market or sold as scrap material). Therefore, implementation questions such as the following arise:

- Should the OEM upgrade to new parts, and if so how fast?
  The OEM may decide to preventively replace all old parts by new parts on the one hand. On the other extreme, he can also determine not to use the new parts at all and stick to old parts by repairing failed old parts. An intermediate approach may be to replace old parts by new ones upon the failure of old parts.
- What should the OEM do with old parts?
  The OEM can decide to keep repairing old parts once they fail, or not to repair old parts because he may want to implement new parts as soon as possible. In addition, old parts can be salvaged, e.g. sold on a second hand market if there is demand for them or sold as scrap material. However, the old parts can also be kept on stock in order to quickly prevent to failures.
- How many new parts should the OEM produce for the transition?
  The number of new parts that are produced determine the investment costs for production, but also determine the uptime of systems and they affect decisions for the old parts that are on stock (salvage or use).
- Should the OEM produce all new parts at once, or should he spread the production over time (and how)?
  Deferring the production is desirable, because this enables the OEM to spread the investment costs, lower associated risk, and it enables him to reduce the discounted costs. Furthermore, postponing the production of (some) new parts may result in a decrease in the production price, as the technology becomes more mature (Hartman and Tan, 2014).

For each of the implementation questions, the OEM has to make a decision, and these decisions are interrelated. This means that he follows a certain implementation strategy. An implementation strategy determines how many new parts the OEM produces and how the new parts are implemented (when and whether an old part is replaced by a new one). Furthermore, an implementation strategy prescribes how many functional old parts from stock are salvaged (sold on a second hand market), since these may become superfluous during the finite horizon. Finally, an implementation strategy also determines whether failed old parts are repaired and used. The objective of the OEM is to select an implementation strategy that results in the highest profit earned over some finite time horizon.

In this chapter, we study four implementation strategies for given production quantities of new parts.

- Stay Put: The OEM does not produce any new parts, repairs the old parts if they fail, and salvages old parts at the end of the horizon.
- Rapid Upgrade: The OEM produces new parts and directly replaces all old parts by new parts. He salvages the old parts immediately (at the start of the horizon). The OEM repairs the new parts once they fail, and he salvages new parts at the end of the horizon.
- Instant Invest: The OEM produces all new parts before the start of the horizon and decides whether to replace a failed part by a new or old one upon each failure. He repairs all new failed parts and old parts are salvaged. Salvaging of old parts occurs during and at the end of the horizon, while salvaging of new parts only occurs at the end of the horizon.
- Phased Invest: The same strategy as Instant Invest, except for the fact that the OEM produces some parts before the horizon and some parts arrive after a number of periods.

We assume – without loss of generality – that it is more profitable to first repair a failed item (either old or new) and salvage the repaired part compared to salvaging the failed part as such. This assumption is only relevant for Instant Invest and Phased Invest.

Under Stay Put, the OEM does not move to the new parts, i.e., no new parts are produced and thus the OEM does not incur the production costs for new parts. However, he cannot reap the benefits of the extra performance that result from upgrading. Rapid Upgrade preventively replaces all old parts by new ones and salvages the old parts. The benefit of this strategy is that the OEM can immediately reap the benefits of the extra performance from the new parts. However, he incurs additional costs of immediate upgrading, because the systems (in which the parts are built) are interrupted during service. The previous two strategies are rather extreme because the OEM either does not move to new parts or he instantaneously moves to the

new parts. Instant Invest, however, produces all new parts at once and gradually transitions to the new parts. Furthermore, Instant Invest assumes that the transition to new parts has been set in motion (e.g. by higher management), and old parts are not used once these are repaired but these are salvaged. Finally, the OEM considers a fourth strategy, Phased Invest. Phased Invest is identical to Instant Invest, except for the fact that it spreads production orders (and costs) over time.

Instant Invest and Phased Invest are complex strategies for which we need advanced mathematical techniques, while Stay Put and Rapid Upgrade are relatively simple strategies that are frequently encountered in practice. Therefore, we focus – in the main body of this chapter – on Instant Invest and Phased Invest, and we consider Stay Put and Rapid Upgrade to be benchmark strategies discussed in Appendix 5.A and 5.B, respectively.

The expected generated profit for each of the implementation strategies is derived under given production quantities of new parts (Rapid Upgrade, Instant Invest, and Phased Invest) and given the arrival time of the second production order (Phased Invest). Next, we optimize the production quantities of new parts and the arrival time of the second production order for the appropriate strategies.

At this point, we remark that there exists a parallel between producing new parts and purchasing these. Our model is also applicable when we consider a company that is offered an upgrade, and this company can decide to purchase new parts. Such companies can be Maintenance, Repair, and Overhaul (MRO) companies like Dutch Railways or KLM Royal Dutch Airlines. Subsequently, the company is interested in the same implementation strategies as discussed above. For the remainder of this paper, we consider an OEM that produces new parts and he has to determine how to implement the upgrades.

Our contributions are the following: we propose a mathematical decision support model, and (i) we present a Markov decision process formulation for Instant Invest wherein the OEM produces the new parts all at once (before the horizon). Secondly, (ii) we show that the optimal decision rule for Instant Invest is not necessarily such that a failed part is replaced by a new part (if available). However, for practical instances we do observe that new parts are used to replace failed ones (if available). Next, (iii) we extend the formulation of Instant Invest to a formulation for Phased Invest. Phased Invest considers two moments at which the OEM can produce new parts, and we show that the formulation can be extended rather easily to an arbitrary number of production moments. Additionally, we illustrate (iv) that Stay Put is optimal if the new component is marginally better than the old component, i.e., the OEM should not replace the old parts by new ones. Furthermore, Rapid Upgrade is optimal when the new component is substantially better than the old one, and thus the OEM should replace old parts by new ones as soon as possible. The more advanced strategies Instant Invest and Phased Invest are useful when this difference between both components less dramatic. Hence, the OEM should gradually replace the old parts by new parts. The expected profit difference between the four strategies

can be notable. Finally, (v) we show that Phased Invest generates strictly higher expected profit than Instant Invest, and this difference can be notable too, but it comes at a high complexity expense with respect to the model formulation and its optimization procedure.

The remainder of this chapter is organized as follows. In Section 5.2 we discuss literature that is related to our problem, and we present our model in Section 5.3. In Section 5.4, we formulate Instant Invest and we analyze this formulation in the same section. In Section 5.5, we extend the formulation for Instant Invest to a formulation for Phased Invest. Finally, we numerically study the various implementation strategies in Section 5.6, and we conclude this chapter in Section 5.7.

5.2. Literature

Our research relates to problems in which units (systems or parts) are replaced due to technological obsolescence. New units become available throughout time and the question is whether to replace the current (old) units by the new units that are available. The literature studying these problems can be decomposed into a single unit stream and a multi–unit stream.

In the single unit stream, the effects of technological change are extensively studied in a finite horizon setting. We restrict this literature overview to technological change that occurs at discrete moments in time, and the arrival of technological change is modeled as a stochastic process, see for example Nair and Hopp (1992), Nair (1995), and Rajagopalan et al. (1998). Their objective is to determine whether to replace a unit by a currently available version or to wait until a newer and better unit is available. Most work in this literature stream neglects the effect that old units from stock have on the replacement decision, except for Nguyen et al. (2013). These authors formulate their problem as a Markov decision process, and they focus on numerical results instead of analytical ones, because the latter are hard to establish due to the complexity of the model. The main difference with our work is that we consider multiple units (we consider multiple parts that each occur once in a system) and we simplify the technological change such that only one new component is available. Furthermore, we include old units that are on stock in our model (Instant Invest and Phased Invest), and are able to provide some preliminary insight in an optimal decision rule.

In a multi–unit setting, some authors study decisions wherein a degraded old unit can be replaced by an 'as good as new' working old unit, i.e., the decision maker cannot implement a new unit with better performance. Authors consider a finite horizon and optimize the decision whether to keep or replace an old unit by using Markov decision theory, see for example Chand et al. (2000), Hartman (2000), and Hsu et al. (2011). We refer to Hartman and Tan (2014) for an extensive literature overview. The first work studying such a replacement problem with multiple units

that operate independently but are economically dependent has been studied by Jones et al. (1991). They study the replacement problem in a setting where units degrade deterministically. Childress and Durango-Cohen (2005) extend the model of Jones et al. (1991) by considering stochastic degradation. However, the previously discussed works do not consider new units that are available, i.e., they neglect the presence of redesigns/upgrades.

Other authors, however, do consider the presence of new units that are available and have better performance. Mercier and Labeau (2004) study a multi–unit setting for a finite horizon in which the authors are interested in the replacement policy for a number of identical old units, when new units have been released. These new units have better performance as they have a lower failure rate and a lower energy consumption. Mercier and Labeau (2004) introduce a K strategy for a number of identical old units, where old units are correctively replaced by new ones until K old units have been replaced. The remaining old units are then preventively replaced. Furthermore, the authors do not consider the temporal aspect of the revenue loss due to idling of units, i.e., the length of the downtime of a unit has no effect on their costs. Mercier (2008) extend the work by Mercier and Labeau (2004) such that general failure rates are incorporated that enable Mercier (2008) to model degradation. The effects that stock units have on the replacement policy are excluded from the analysis in Mercier and Labeau (2004), and Mercier (2008). Öner et al. (2015) do consider units on stock, although only for the new units. The authors simplify the K strategy to obtain analytical insights in the relationship between the stock units and the replacement policy. Hence, the authors consider: (i) preventively replacing all old units by new ones at time 0 (like Rapid Upgrade), and (ii) correctively replacing old units by new units. Under the second policy, the decision maker decides how many new units q_0 to order at time 0, and how large the constant order size of new units is after time 0. The authors are interested in deriving an efficient procedure to determine the optimal solution of q_0 for their problem. However, Öner et al. (2015) do not consider the profit losses that are incurred when systems are idle because there are no units on stock. Furthermore, they do not model the effect that old units from stock have on the replacement decision. This effect of old units on stock (and new units on stock) is studied in Clavareau and Labeau (2009) in a simulation model. We, on the other hand, model our problem by using Markov decision theory, and are able to provide some more grip on the structure of an optimal decision rule.

Finally, our work relates to literature on repairable unit inventory systems, because we repair new units and keep some new units on stock. The vast majority of literature typically assumes that the number of installed units is large and that the demand process is independent on the number of units in repair, see for example the books by Muckstadt (2005) and van Houtum and Kranenburg (2015). A small literature stream studies such problems with a small installed base so that the demand process depends on the number of units in repair, see Gross and Ince (1978). The authors consider an infinite horizon model, and they are primarily interested in determining the number of units on stock needed to minimize the costs of their model. We also

consider a problem where the demand process depends on the number of operating units. However, we have a finite horizon problem wherein old units can be replaced by new ones (that generate more revenue) and we are interested in an optimal decision rule that determines whether to replace a failed unit by an old or new failed and how many old units from stock to salvage. We provide an overview of the most related papers in Table 5.1.

Paper	Multiple units	Improved new units	Old parts from stock	Losses due to downtime
Nair (1995)		x		
Childress and Durango-Cohen (2005)	x			
Nguyen et al. (2013)		x	x	
Mercier (2008)	x	x		
Clavareau and Labeau (2009)	x	x	x	
Öner et al. (2015)	x	x		
This chapter	x	x	x	x

Table 5.1: Comparison of most related papers

5.3. Model

We consider an OEM that is responsible for the operation of N systems due service contracts that he closed with his customers (for instance, a performance based contract). We assume that the OEM has closed identical service contracts with each of his customers, and that it is profitable for the OEM and the customer to use such a service contract. The N systems are used for a remaining finite lifetime of n periods (days) and the OEM considers a periodic discount factor $0 \leq \alpha \leq 1$. We direct our attention to critical repairable parts that occur once in a system. This assumption is easily generalizable: if a part occurs x times in a system, then we consider xN installed parts. In the remainder, we work with parts that occur once in a system, and we refer to the installed parts N as the installed base. The OEM believes that the parts currently used are underperforming. Therefore, he considers to design a new component with better performance in terms of the failure rate and the revenue that the part generates (e.g. increased productivity). If the OEM develops and produces

the new parts, he pays unit production cost τ. Furthermore, we assume that the OEM decides before the horizon that he produces m_1 new parts prior to the horizon and m_2 new parts that arrive in period L. The production decisions are made prior to the horizon, because the OEM closes contracts (before the start of the horizon) that state when the new parts are delivered. The production costs are paid upon arrival of the new parts.

In addition to the production of new parts, the OEM has s old parts on stock at time 0. An operating old part generates revenue ρ_0 per period, and working old parts can be salvaged during the horizon at unit salvage value w_0, e.g. the parts are sold on a second–hand market or are sold as scrap material. All old parts fail independently, and the failure time for each old part is geometrically distributed with failure probability λ_0 per period. Failed parts are repaired by a repair shop with ample capacity. The OEM pays repair costs $z > 0$ per failure of a part. The repair costs z not only capture the costs of repairing the failed part, but also incorporate costs due to the downtime of a system, e.g. the costs of sending a maintenance engineer, the downtime due to the engineer working on the system, and so on. Furthermore, we assume that the expected repair costs per period are less than the gross revenue, i.e., $\rho_0 \geq z\lambda_0$ (otherwise the OEM is going out of business). The repair time per part is geometrically distributed with repair probability μ per period, and we denote the random variable for the repair time by Y. Furthermore, we assume that it is worthwhile to repair a failed part before salvaging it. We use these properties of the repair process to simplify our formulations (simpler state spaces): if the OEM salvages the failed old part after repair, he earns an expected discounted salvage value per failed old part of $\hat{w}_0 = w_0\mathbb{E}[\alpha^Y]$ upon the start of the repair. This means that we do not have to record the number of old parts in repair, if these old parts are salvaged after repair.

Each operating new part has a higher performance compared to the old part. This implies that a new part generates revenue $\rho_1 \geq \rho_0$ per period. New parts fail independently and are repaired upon failure. The failure time of a new part is geometrically distributed with failure probability $\lambda_1 \leq \lambda_0$ per period. We assume that the repair costs z and the repair process of new parts are identical to those of old parts, for the ease of exposition. However, this assumption can easily be relaxed. Furthermore, new parts can be salvaged after the horizon, either working or failed. If a working new part is salvaged, the OEM earns salvage value $w_1 \geq w_0$. In case a failed new part is salvaged, he uses the same expected discounted salvage value method as before. That is, the OEM immediately earns $\hat{w}_1 = w_1\mathbb{E}[\alpha^Y]$ once the repair of the failed new part starts.

We assume that at most one event occurs in a period. Such an event is either a failure of a part or a repair of a part. Furthermore, the assumption of at most one event per period is reasonable if the length of a period is sufficiently short (say a day), or if the failure rate of parts is low, which is often the case in industries using capital intensive systems (Sherbrooke, 2004; van Houtum and Kranenburg, 2015). The sequence of events in a period x is given as follows. First, new parts arrive in a period. Then, the

OEM earns the revenue generated by the working old and new parts at the start of period x. Subsequently, a number of old parts from stock is salvaged and the OEM earns a salvage value w_0 per old part. Next, an event may occur (at most one by assumption): a failure of a part (old or new) or the repair of a part. If a failure occurs, the OEM incurs the repair costs and replaces the failed part by an old part from stock, by a new part, or not at all. We assume that the latter decision is only selected in case there are insufficient old and new parts available[1]. Furthermore, the OEM earns the expected discounted salvage value $\hat{w}_0$ ($\hat{w}_1$) if the failed part is old (new) and immediately salvaged after repair. When a repair of a part finishes and the part is not salvaged, it is sent to the stockpoint. This repaired part can only be used to replace a failed part in the next period. Finally, the period ends and we transition to the next state.

The objective is to determine what actions to take in each period, such that we maximize the total discounted expected profit earned over periods $1, 2, \ldots, n$, with periodic discount factor $0 \leq \alpha \leq 1$. The problem is a discounted, finite horizon, discrete time Markov decision process (DTMDP). We make the following remark on the asymptotic behavior of our DTMDP.

Remark 5.1 Our DTMDP is asymptotically equivalent to a discounted, finite horizon, continuous time Markov decision process (CTMDP) with horizon $[0, T]$, in which the failure times and repair times are exponentially distributed rather than geometrically. The asymptotical equivalence follows as $h = T/n$ tends to 0, and the one step transition probabilities are given by hq, where q corresponds to the transition rate (Martin-Löf, 1967).

5.4. Instant Invest

Instant Invest is such that all new parts m are purchased prior to the horizon, i.e., $m_1 = m$ and $m_2 = 0$. These m parts arrive in period 1, before any event or action has taken place. This is equivalent to receiving all m new parts prior to the horizon, but paying for these in the first period. We will use this equivalent representation in the remainder of this section. The state space consists of a number of dimensions: the number of new parts in repair $0 \leq X_1 \leq m$, the number of operating new parts $0 \leq N_1 \leq m$, the number of new parts on stock $0 \leq s_1 \leq m$, the number of operating old parts $0 \leq N_0 \leq N$, and the number of old parts on stock $0 \leq s_0 \leq s$. We do not consider the number of old parts in repair X_0, because the old parts leave the system once they have failed. The foregoing implies a five dimensional state space. However, we reduce this to a three dimensional state space. The number of new parts (either in repair or operating) is given by $N - N_0$, and the number of new parts that are on stock

[1] In our motivational setting, non–operating systems cause large revenue losses that force decision makers to prevent system idling. As a consequence, old parts are used to prevent idling of systems; i.e., if no new parts are available upon the failure of a part, the decision maker installs an old part.

or operating is given by $m - X_1$. Consequently, the number of operating new parts $N_1 = \max\{N - N_0, m - X_1\}$ is derived from N, N_0, m, X_1. Furthermore, the number of new parts on stock is given by $s_1 = m - N_1 - X_1 = m - X_1 - \max\{N - N_0, m - X_1\}$. Hence, the state space dimension is reduced from five to three by using the production quantity m. Therefore, the state space is defined by $\mathcal{S}_m = \{(X_1, N_0, s_0) : 0 \leq X_1 \leq m, 0 \leq N_0 \leq N, 0 \leq s_0 \leq s\}$, where the subscript m is added because the production quantity m is given. We refer to a state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$, where i, j, k correspond to X_1, N_0, and s_0, respectively.

At the start of each period $1 \leq x \leq n$ the OEM takes an action. Let $\mathcal{A}_m(\mathfrak{s})$ be the action space given the production quantity m and state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$. Each action $a \in \mathcal{A}_m(\mathfrak{s})$ is two dimensional. First, the OEM chooses whether to replace a failed part by an old part; by a new part; or not at all. We assume that the latter decision is only selected in case there are insufficient old and new parts available at the start of the period. The second dimension of an action describes how many old parts from stock to salvage. We denote this quantity by $0 \leq v \leq k$. Hence, an action $a \in \mathcal{A}_m(\mathfrak{s})$, given state $\mathfrak{s} = (i, j, k)$ is characterized as a tuple $a = (u, v)$. If $u = 0$, a failed part is replaced by an old part; if $u = 1$ a failed part is replaced by a new one; and if $u = 2$ the OEM does nothing, because there are no old parts on stock and there are insufficient new parts at the start of the period. We note that u prescribes the replacement action that occurs if a failure occurs. If no failure occurs, no failed part needs replacement and thus the decision is redundant. We make a case distinction for the definition of $\mathcal{A}_m(\mathfrak{s})$. In the first case, there are only old parts on stock at the start of the period; the OEM salvages v old parts from stock; and he replaces a failed part by an old one (if possible after salvaging). For the second case, there are no old parts on stock and there are sufficient new parts available at the start of the period; the OEM salvages $v = 0$ old parts from stock; and he replaces a failed part by a new part (if possible after salvaging). In the third case, there are old parts on stock and sufficient new parts are available at the start of the period; the OEM salvages v old parts (from stock) and he can choose whether to replace a failed part by an old or new part. The last case has no old and new parts, and thus a failed part cannot be replaced.

$$\mathcal{A}_m(\mathfrak{s}) = \begin{cases} \{(0, v) : 0 \leq v \leq k\} & \text{if } k \geq 1, m \leq N - j + i \\ \{(1, v) : v = 0\} & \text{if } k = 0, m > N - j + i \\ \{(u, v) : u \in \{0, 1\}, 0 \leq v \leq k\} & \text{if } k \geq 1, m > N - j + i \\ \{(2, 0)\} & \text{otherwise.} \end{cases}$$

Next, we consider the one step transition probability from state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$ to state $\mathfrak{s}' \in \mathcal{S}_m$ under action $a = (u, v) \in \mathcal{A}_m(\mathfrak{s})$. Let $p_m(\mathfrak{s}'|\mathfrak{s}, a)$ denote this one step transition probability, and it is given for each $\mathfrak{s} \in \mathcal{S}_m$ and all $a \in \mathcal{A}_m(\mathfrak{s})$ by Eq. (5.1).

Eq. (5.1a) represents the case that $v < k$ old parts from stock are salvaged and that a failed old part is replaced by an old part from stock. Eq. (5.1b) covers the case that all old stock parts $v = k$ are salvaged and subsequently an old part fails. Then, the

failed part cannot be replaced by an old part from stock and thus the installed base of old parts decreases. The expression in Eq. (5.1c) corresponds to replacing a failed old part by a new part, and salvaging v old parts from stock. The number of operating old parts decreases by one, while the number of new parts in repair does not increase. Eq. (5.1d) represents that the failed old part cannot be replaced, because there are no old stock parts and there are insufficient new parts. Thus, only the number of operating old parts decreases with one.

$$p_m(\mathfrak{s}'|\mathfrak{s},a) = \begin{cases} j\lambda_0 & \text{if } \mathfrak{s}' = (i,j,k-1-v), u=0, v<k, \text{(5.1a)} \\ j\lambda_0 & \text{if } \mathfrak{s}' = (i,j-1,0), u=0, v=k, \text{(5.1b)} \\ j\lambda_0 & \text{if } \mathfrak{s}' = (i,j-1,k-v), u=1, \text{(5.1c)} \\ j\lambda_0 & \text{if } \mathfrak{s}' = (i,j-1,0), u=2, \text{(5.1d)} \\ (N-j)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j+1,k-1-v), \\ & \quad u=0, v<k, m>N-j+i, \text{(5.1e)} \\ (N-j)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j,0), \\ & \quad u=0, v=k, m>N-j+i, \text{(5.1f)} \\ (m-i)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j+1,k-1-v), \\ & \quad u=0, v<k, m\leq N-j+i, \text{(5.1g)} \\ (m-i)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j,0), \\ & \quad u=0, v=k, m\leq N-j+i, \text{(5.1h)} \\ (N-j)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j,k-v), u=1, \text{(5.1i)} \\ (m-i)\lambda_1 & \text{if } \mathfrak{s}' = (i+1,j,0), u=2, \text{(5.1j)} \\ i\mu & \text{if } \mathfrak{s}' = (i-1,j,k-v), \text{(5.1k)} \\ 1 - \sum_{\hat{\mathfrak{s}}\in\mathcal{S}_m:\hat{\mathfrak{s}}\neq\mathfrak{s}'} p_m(\hat{\mathfrak{s}}|\mathfrak{s},a) & \text{if } \mathfrak{s}' = (i,j,k-v), \text{(5.1l)} \\ 0 & \text{otherwise.} \text{(5.1m)} \end{cases}$$

The expression in Eq. (5.1e) represents the case wherein there are sufficient new parts such that $N-j$ new parts are operating, $v<k$ old parts from stock are salvaged, and a failed *new* part is replaced by an old stock part. We obtain Eq. (5.1f) if the OEM salvages all $v=k$ old parts from stock and he cannot replace a failed new part. The expression in Eq. (5.1g) considers the same action as Eq. (5.1e), but there are *insufficient* new parts and consequently $m-i$ new parts are operating. Eq. (5.1h) describes that there are insufficient new parts and all $v=k$ old parts from stock are salvaged. Thus, the failed new part is not replaced.

We obtain Eq. (5.1i) if the OEM replaces a failed old part by a new part and he salvages v old stock parts. Finally, a failed new part is not replaced when the OEM runs out of old parts on stock and has insufficient new parts available; this yields Eq. (5.1j). Eq. (5.1k) covers the repair probability.

The expressions Eq. (5.1a)–(5.1k) are the transition probabilities for a failure or repair event. If no repair or failure occurs, we move from state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$ to state $\mathfrak{s}' = (i, j, k - v) \in \mathcal{S}_m$, because only v old parts from stock are salvaged. Thus, we obtain Eq. (5.1l).

The OEM's objective is to study the total expected discounted profit over n periods of the DTMDP, with a periodic discount factor $0 \leq \alpha \leq 1$. Each state $\mathfrak{s} \in \mathcal{S}_m$ generates revenue $\rho_m(\mathfrak{s})$ per period. Moreover, if a failure occurs, the OEM incurs repair costs z per repair (and he repairs each failed part), independent of whether the failed part is old or new. We define $r_m(\mathfrak{s})$ as the expected repair costs in a period given state $\mathfrak{s} \in \mathcal{S}_m$. Only the state determines the expected repair costs, because the salvage decision in period x does not affect the failure or repair that may occur in x. The OEM also earns a salvage value w_0 per working old part that is salvaged, and he earns $\hat{w}_0$ for salvaging a failed old part. Therefore, the expected salvage value in a period depends on the action a (the salvage quantity of working old parts) and on the state $\mathfrak{s}$ (a failed old part is salvaged). New parts are not salvaged during the horizon, and thus we discuss the salvaging of new parts when we consider the terminal value of our DTMDP (at the end of this section). Let us denote the expected salvage value in a period by $c(\mathfrak{s}, a)$ given action a and state $\mathfrak{s}$. We use $\rho_m(\mathfrak{s})$, $r_m(\mathfrak{s})$, and $c(\mathfrak{s}, a)$ to determine the total expected profit earned in a period

$$\sigma_m(\mathfrak{s}, a) = \rho_m(\mathfrak{s}) - r_m(\mathfrak{s}) + c(\mathfrak{s}, a),$$

given state $\mathfrak{s}$ and action a. Let us first study the revenue $\rho_m(\mathfrak{s})$ in a state $\mathfrak{s} \in \mathcal{S}_m$, which depends on the number of old and new parts that are operating. This number of parts can be derived from the state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$:

$$\rho_m(\mathfrak{s}) = \begin{cases} j\rho_0 + (m - i)\rho_1 & \text{if } m \leq N - j + i \\ j\rho_0 + (N - j)\rho_1 & \text{otherwise.} \end{cases}$$

The first term follows directly from having j operating old parts. For the second term, there are two cases: the first represents the revenue earned when the OEM has insufficient new parts, whereas the second case denotes the revenue earned when the OEM has sufficient new parts.

The expected repair costs $r_m(\mathfrak{s})$ for a given state $\mathfrak{s}$ are as follows. Each part that fails is repaired (independent on whether it is old or new) and consequently the OEM incurs the repair costs z. The probability that a part fails depends on the number of operating old and new parts. These numbers of operating old and new parts can be derived from the state $\mathfrak{s}$. Hence, we obtain for the expected repair costs

$$r_m(\mathfrak{s}) = \begin{cases} z\,(j\lambda_0 + (m - i)\lambda_1) & \text{if } m \leq N - j + i \\ z\,(j\lambda_0 + (N - j)\lambda_1) & \text{otherwise.} \end{cases}$$

For a given action $a = (u, v) \in \mathcal{A}_m(\mathfrak{s})$, the OEM salvages v old stock parts at unit salvage value w_0. Hence, he earns $w_0 v$. Furthermore, the OEM earns salvage

value for a failed old part $\hat{w}_0 = w_0\mathbb{E}[\alpha^Y]$, where $\mathbb{E}[\alpha^Y]$ is the generating function of the random variable Y evaluated at α. The generating function of Y is given by $\mathbb{E}[\alpha^Y] = \sum_{n=1}^{\infty}(1-\mu)^{n-1}\mu\alpha^n = \frac{\alpha\mu}{1-\alpha(1-\mu)}$, where Y is geometrically distributed on the support $\{1, 2, \ldots\}$. Hence, the OEM earns $\hat{w}_0 = w_0\frac{\alpha\mu}{1-\alpha(1-\mu)}$ if an old part fails. The probability that an old part fails in period x is $j\lambda_0$. Therefore, we obtain for the expected salvage value

$$c(\mathfrak{s}, a) = j\lambda_0\hat{w}_0 + w_0 v.$$

Now, we can determine the expected profit $\sigma_m(\mathfrak{s}, a)$ for a given state $\mathfrak{s} \in \mathcal{S}_m$ and action $a \in \mathcal{A}_m(\mathfrak{s})$. We use this expression – along with the state space, action space and one step transition probabilities – to obtain the recursion that determines the optimal decision rule maximizing the total expected profit earned over n periods, with periodic discount factor $0 \leq \alpha \leq 1$. Let $V_x(\mathfrak{s}|m)$ be the maximum total expected discounted profit earned over periods $x, x+1, \ldots, n$, given the production of m new parts and given that we are in state $\mathfrak{s} \in \mathcal{S}_m$ in period x. The recursion for $V_x(\mathfrak{s})$ for all $1 \leq x \leq n$ is given by

$$V_x(\mathfrak{s}|m) = \max_{a\in\mathcal{A}_m(\mathfrak{s})}\left\{\sigma_m(\mathfrak{s}, a) + \alpha\sum_{\mathfrak{s}'\in\mathcal{S}_m} p_m(\mathfrak{s}'|\mathfrak{s}, a)V_{x+1}(\mathfrak{s}'|m)\right\}, \tag{5.2}$$

and the optimal decision rule for all periods $1 \leq x \leq n$ is determined by

$$\pi_x(\mathfrak{s}|m) = \operatorname*{argmax}_{a\in\mathcal{A}_m(\mathfrak{s})}\left\{\sigma_m(\mathfrak{s}, a) + \alpha\sum_{\mathfrak{s}'\in\mathcal{S}_m} p_m(\mathfrak{s}'|\mathfrak{s}, a)V_{x+1}(\mathfrak{s}'|m)\right\}.$$

After the final period n, the OEM earns a terminal reward due to the salvaging of the remaining operating old parts at w_0, and due to the salvaging of all m new parts. Suppose we are in state $\mathfrak{s} = (i, j, k) \in \mathcal{S}_m$ at the end of the horizon n. Then, $j + k$ working old parts are salvaged at the unit salvage value w_0. Furthermore, all operating new parts $m - i$ are salvaged at w_1. The i new parts in the repair shop are repaired and subsequently salvaged. Hence, the OEM earns the discounted salvage value of $\hat{w}_1$ for these i parts. Overall, he earns an expected terminal reward of $V_{n+1}(\mathfrak{s}|m) = (j + k)w_0 + (m - i)w_1 + i\hat{w}_1$ in state $\mathfrak{s} \in \mathcal{S}_m$.

Next, we consider the optimal decision rule for Instant Invest. We expect that – under an optimal decision rule – the use of new parts is preferred over the use of old parts. That is, if the OEM does not salvage all old parts on stock, and there are new parts available, we expect that the use of new parts is preferred over the use of old parts. However, this is not necessarily true as Example 5.1 illustrates.

Example 5.1 Let $n = 2$, $N = 1$, $m = 1$, $s_0 = 1$, and consider the initial state $(0, 1, 1)$. The replacement in the second period is irrelevant, because the horizon ends after this period. Hence, we are only concerned with the replacement in period 1

and with the salvage quantities in periods 1 and 2. Let $G^1(a,b)$ be the expected revenue earned over the horizon when the OEM replaces a failed part by a new part and salvages a and b old parts from stock in period 1 and 2, respectively. Similarly, $G^0(a,b)$ is the expected revenue earned over the horizon when he replaces a failed part by an old one and salvages a and b old parts from stock in period 1 and 2, respectively. Then, we have the following costs for each of the feasible decision rules:

$$\begin{aligned}
G^0(0,0) &= \lambda_0^2(\rho_0 + \alpha\rho_0 - z - \alpha z + \hat{w}_0 + \alpha\hat{w}_0) \\
&\quad +\lambda_0(1-\lambda_0)(\rho_0 + \alpha\rho_0 - z + \alpha^2 w_0 + \hat{w}_0) \\
&\quad +(1-\lambda_0)\lambda_0(\rho_0 + \alpha\rho_0 - z\alpha + \alpha^2 w_0 + \alpha\hat{w}_0) \\
&\quad +(1-\lambda_0)^2(\rho_0 + \alpha\rho_0 + 2\alpha^2 w_0) + \alpha^2 w_1 \\
&= \rho_0 + \alpha\rho_0 - \lambda_0 z(1+\alpha) + \lambda_0\hat{w}_0(1+\alpha) + (1-\lambda_0)2\alpha^2 w_0 + \alpha^2 w_1,
\end{aligned}$$

$$\begin{aligned}
G^0(0,1) &= \lambda_0(\rho_0 - z + \hat{w}_0 + \alpha w_0) + (1-\lambda_0)\lambda_0(\rho_0 + \alpha\rho_0 - z\alpha + \alpha w_0 + \alpha\hat{w}_0) \\
&\quad +(1-\lambda_0)^2(\rho_0 + \alpha\rho_0 + \alpha w_0 + \alpha^2 w_0) + \alpha^2 w_1 \\
&= \rho_0 + (1-\lambda_0)\alpha\rho_0 - z\lambda_0(1+\alpha(1-\lambda_0)) + \lambda_0\hat{w}_0(1+\alpha(1-\lambda_0)) \\
&\quad +\alpha w_0(1+\alpha(1-\lambda_0)^2) + \alpha^2 w_1,
\end{aligned}$$

$$\begin{aligned}
G^1(0,0) &= \lambda_0\lambda_1(\rho_0 + \alpha\rho_1 - z - \alpha z + \hat{w}_0 + \alpha^2\hat{w}_1) \\
&\quad +\lambda_0(1-\lambda_1)(\rho_0 + \alpha\rho_1 - z + \alpha^2 w_1 + \hat{w}_0) \\
&\quad +(1-\lambda_0)\lambda_0(\rho_0 + \alpha\rho_0 - z\alpha + \alpha^2 w_1 + \alpha\hat{w}_0) \\
&\quad +(1-\lambda_0)^2(\rho_0 + \alpha\rho_0 + \alpha^2 w_0 + \alpha^2 w_1) + \alpha^2 w_0 \\
&= \rho_0 + \alpha\rho_1\lambda_0 + \alpha\rho_0(1-\lambda_0) - \lambda_0 z - \lambda_0 z\alpha - \lambda_0\lambda_1\alpha z + \lambda_0^2 z\alpha + \hat{w}_0\lambda_0 \\
&\quad +\lambda_0\lambda_1\alpha^2\hat{w}_1 + (1-\lambda_0\lambda_1)\alpha^2 w_1 + \lambda_0\alpha\hat{w}_0 - \lambda_0^2\hat{w}_0\alpha + \alpha^2 w_0 - 2\lambda_0\alpha^2 w_0 \\
&\quad +\lambda_0^2\alpha^2 w_0 + \alpha^2 w_0 \\
&= \rho_0 + \alpha\rho_0 - \lambda_0 z(1+\alpha) + \lambda_0\hat{w}_0(1+\alpha) + (1-\lambda_0)2\alpha^2 w_0 + \alpha^2 w_1 \\
&\quad +\alpha\lambda_0(\rho_1 - \rho_0) - \lambda_0\alpha z(\lambda_1 - \lambda_0) + \lambda_0\lambda_1\alpha^2(\hat{w}_1 - w_1) - \lambda_0^2\alpha(\hat{w}_0 - w_0\alpha) \\
&= G^0(0,0) + \alpha\lambda_0(\rho_1 - \rho_0) - \lambda_0\alpha z(\lambda_1 - \lambda_0) + \lambda_0\lambda_1\alpha^2(\hat{w}_1 - w_1) \\
&\quad -\lambda_0^2\alpha(\hat{w}_0 - w_0\alpha).
\end{aligned}$$

The decision rule where the OEM salvages all old parts from stock in the first period, and replaces a failed part by an old part from stock is infeasible. Hence, $G^0(1,0)$ is not defined. Furthermore, if the OEM replaces a failed part by a new part and the horizon consists of only two periods, he does not use the old part from stock. Hence, we have $G^1(0,1) = G^1(0,0) - \alpha^2 w_0 + \alpha w_0$ and $G^1(1,0) = G^1(0,0) - \alpha^2 w_0 + w_0$. Now, let $\lambda_0 = \lambda_1 = 0.03$, $\mu = 0.001$, $\alpha = 0.9$, $w_1 = 1.1$, $w_0 = 0.001$, $z = 1$, $\rho_1 = 0.05$ and $\rho_0 = 0.04$. This yields $G^1(0,0) < G^1(0,1) < G^1(1,0) < G^0(0,1) < G^0(0,0)$ and thus the optimal action is *not* to install a new part, but rather to install an old one. $\diamond$

Example 5.1 installs an old part rather than a new part, because of the end–of–horizon effect of salvaging. Instant Invest does not take the risk of losing salvage value $w_1 - \hat{w}_1$

because this salvage value is relatively high compared to the expected revenue that is generated. However, if we consider practical instances, we typically have a long horizon and discounting. Furthermore, the failure rates of parts are typically low in capital goods settings, and the revenue that is generated per time unit is also substantial. As a consequence, the end–of–horizon effects are relatively small for such instances, and the behavior of preferring old parts over new parts hardly occurs in practice. Thus, we propose to consider a simpler formulation for Instant Invest in the remainder of this chapter, in which the OEM installs new parts if these are available; he installs old ones only if he has old parts on stock and no new parts available; and he idles only if there are no old or new parts available. This yields a simplified action space $\mathscr{A}_m(\mathfrak{s})$ given m and state $\mathfrak{s} \in \mathcal{S}_m$, in which we do not need to optimize the decision of installing an old, new, or no part upon a failure:

$$\mathscr{A}_m(\mathfrak{s}) = \begin{cases} \{(0,v) : 0 \leq v \leq k\} & \text{if } k \geq 1, m \leq N-j+i \\ \{(1,v) : 0 \leq v \leq k\} & \text{if } k \geq 0, m > N-j+i \\ \{(2,0)\} & \text{otherwise.} \end{cases} \tag{5.3}$$

We also obtain fewer non–zero one step transition probabilities. That is, for a given state $\mathfrak{s} \in \mathcal{S}_m$ and action $a \in \mathscr{A}_m(\mathfrak{s})$ the one step transition probability $p_m(\mathfrak{s}'|\mathfrak{s}, a)$ is defined by Equations (5.1a)–(5.1d) and Equations (5.1g)–(5.1m). A major practical consequence simplifying the replacement action is that we are able to solve problems with the same computation time but where the state space has twice as many states compared to using $\mathcal{A}_m(\mathfrak{s})$. This results from the fact that the action space decreases by a factor two (we no longer need to differentiate between replacement actions). Hence, practical problems can still be solved within reasonable amounts of time, which would otherwise be significantly harder.

The OEM is also interested in the optimal production quantity of new parts m^*. Thus, we study the behavior of the expected profit $V_1(0, N, s|m) - m\tau$ with respect to m, given that our starting state is $(0, N, s)$. Therefore, we let $\mathcal{V}(\mathfrak{s}, m) = V_1(\mathfrak{s}|m) - m\tau$ for each $\mathfrak{s} \in \mathcal{S}_m$, and we are particularly interested in $\mathcal{V}(0, N, s, m)$ in the remainder. We can determine a compact interval $\{0, 1, \ldots, \overline{m}\}$ that contains m^*, i.e., we can derive an upper bound $\overline{m}$ for m^*. We do this by considering an upper bound to $\mathcal{V}(0, N, s, m)$.

Proposition 5.1 *Given $m \in \mathbb{N}$ and state $\mathfrak{s} \in \mathcal{S}_m$, we have*

$$(s_0 + N)w_0 + (\rho_1 - z\lambda_1)N \sum_{i=1}^{n} \alpha^{i-1} - m\left(\tau - w_1\alpha^n\right) \geq \mathcal{V}(0, N, s, m).$$

Proof. Let $m \in \mathbb{N}$ and $\mathfrak{s} \in \mathcal{S}_m$, then we have $\rho_m(\mathfrak{s}) - r_m(\mathfrak{s}) \leq (\rho_1 - z\lambda_1)N$. The inequality follows as $\rho_1 \geq \rho_0 \geq z\lambda_0 \geq z\lambda_1$ since $\rho_1 \geq \rho_0$, $\lambda_1 \leq \lambda_0$ and $\rho_0 \geq z\lambda_0$. This means that new parts generate the most revenue per period, and $(\rho_1 - z\lambda_1)N$

represents that all working parts are new. If we take discounting into account for periods $1, 2, \ldots, n$ we find the following upper bound for the revenue and repair costs, $(\rho_1 - z\lambda_1)\sum_{i=1}^{n} \alpha^{i-1}$.

If we consider the salvage value, the OEM salvages at most $s_0 + N$ old parts for which he either earns w_0 or $\hat{w}_0 \leq w_0$ salvage value. Furthermore, the salvage value depends on period x due to discounting. Hence, the OEM earns strictly less total discounted salvage value than $(s_0 + N)w_0$.

Moreover, the OEM produces m parts at unit price τ and he salvages these m parts after the horizon. The salvage value that he earns from salvaging a new part is at most w_1. Hence, for each produced new part the OEM pays τ and earns at most $w_1\alpha^n$ due to discounting. Therefore, the OEM pays at least a net production cost of $\tau - w_1\alpha^n$ per new part. Combining all the above yields the desired result. □

Since $\rho_1 \geq z\lambda_1$, $\rho_0 \geq z\lambda_0$, and $w_1 \geq 0$, the OEM earns value in every period, i.e., $\sigma_m(\mathfrak{s}, a) \geq 0$ for all $\mathfrak{s} \in \mathcal{S}_m$ and $a \in \mathscr{A}_m(\mathfrak{s})$. Therefore, there exists an $m \in \{1, \ldots, \overline{m}\}$ such that $\mathcal{V}(\mathfrak{s}, m) \geq 0$ for all states $\mathfrak{s} \in \mathcal{S}_m$. Hence, we use Proposition 5.1 to derive an upper bound for the optimal number of new parts to produce by finding an $\overline{m}$ such that all $m > \overline{m}$ yield a non–positive profit:

$$\begin{aligned}\overline{m} &= \left\lfloor \frac{(s_0 + N)w_0 + (\rho_1 - z\lambda_1)N\sum_{i=1}^{n} \alpha^{i-1}}{\tau - w_1\alpha^n} \right\rfloor \\ &= \left\lfloor \frac{(s_0 + N)w_0 + (\rho_1 - z\lambda_1)N(1-\alpha^n)}{(\tau - w_1\alpha^n)(1-\alpha)} \right\rfloor.\end{aligned}$$

This upper bound $\overline{m}$ is loose and the state space increases linearly in m. Thus, enumerating $\mathcal{V}(0, N, s, m)$ for each $m \in \{1, \ldots, \overline{m}\}$ is computationally prohibitive, and may even result in instances too large to fit in 16GB computer memory. An alternative to this enumeration is to explore the value function $V_1(0, N, s|m)$. This $V_1(0, N, s|m)$ is not concavely increasing in m for all instances, see for instance Example 5.2.

Example 5.2 Let $n = 3$, $N = 1$, $s_0 = 0$, and consider the initial state $(0, 1, 0)$ in which we have one installed old part. Then, we have

$$\begin{aligned}V_1(0,1,0|1) = {} & \lambda_0\lambda_1\mu(\rho_0 + \alpha\rho_1 - z - \alpha z + \alpha^3 w_1 + \hat{w}_0) \\ & +\lambda_0\lambda_1(1-\mu)(\rho_0 + \alpha\rho_1 - z - \alpha z + \alpha^3\hat{w}_1 + \hat{w}_0) \\ & +\lambda_0(1-\lambda_1)\lambda_1(\rho_0 + \alpha\rho_1 + \alpha^2\rho_1 - z - \alpha^2 z + \hat{w}_0 + \alpha^3\hat{w}_1) \\ & +\lambda_0(1-\lambda_1)^2(\rho_0 + \alpha\rho_1 + \alpha^2\rho_1 - z + \hat{w}_0 + \alpha^3 w_1) \\ & +(1-\lambda_0)\lambda_0\lambda_1(\rho_1 + \alpha\rho_0 + \alpha^2\rho_1 - z\alpha - z\alpha^2 + \alpha\hat{w}_0 + \alpha^3\hat{w}_1) \\ & +(1-\lambda_0)\lambda_0(1-\lambda_1)(\rho_0 + \alpha\rho_0 + \alpha^2\rho_1 - z\alpha + \alpha\hat{w}_0 + \alpha^3 w_1) \\ & +(1-\lambda_0)^2\lambda_0(\rho_0 + \alpha\rho_0 + \alpha^2\rho_0 - z\alpha^2 + \alpha^3 w_1 + \alpha^2\hat{w}_0) \\ & +(1-\lambda_0)^3(\rho_0 + \alpha\rho_0 + \alpha^2\rho_0 + \alpha^3 w_0 + \alpha^3 w_1),\end{aligned}$$

$$
\begin{aligned}
V_1(0,1,0|2) = {} & \lambda_0\lambda_1\mu(\rho_0 + \alpha\rho_1 - z - \alpha z + \alpha^3 w_1 + \hat{w}_0 + \alpha^2\rho_1 + \alpha^3 w_1) \\
& + \lambda_0\lambda_1^2(\rho_0 + \alpha\rho_1 - z - \alpha z + \alpha^3\hat{w}_1 + \hat{w}_0 + \alpha^2\rho_1 - \alpha^2 z + \alpha^3\hat{w}_1) \\
& + \lambda_0\lambda_1(1-\mu-\lambda_1)(\rho_0 + \alpha\rho_1 - z - \alpha z + \alpha^3\hat{w}_1 + \hat{w}_0 + \alpha^2\rho_1 + \alpha^3 w_1) \\
& + \lambda_0(1-\lambda_1)\lambda_1(\rho_0 + \alpha\rho_1 + \alpha^2\rho_1 - z - \alpha^2 z + \hat{w}_0 + \alpha^3\hat{w}_1 + \alpha^3 w_1) \\
& + \lambda_0(1-\lambda_1)^2(\rho_0 + \alpha\rho_1 + \alpha^2\rho_1 - z + \hat{w}_0 + \alpha^3 w_1 + \alpha^3 w_1) \\
& + (1-\lambda_0)\lambda_0\lambda_1(\rho_0 + \alpha\rho_0 + \alpha^2\rho_1 - z\alpha - z\alpha^2 + \alpha\hat{w}_0 + \alpha^3\hat{w}_1 + \alpha^3 w_1) \\
& + (1-\lambda_0)\lambda_0(1-\lambda_1)(\rho_0 + \alpha\rho_0 + \alpha^2\rho_1 - z\alpha + \alpha\hat{w}_0 + \alpha^3 w_1 + \alpha^3 w_1) \\
& + (1-\lambda_0)^2\lambda_0(\rho_0 + \alpha\rho_0 + \alpha^2\rho_0 - z\alpha^2 + \alpha^3 w_1 + \alpha^2\hat{w}_0 + \alpha^3 w_1) \\
& + (1-\lambda_0)^3(\rho_0 + \alpha\rho_0 + \alpha^2\rho_0 + \alpha^3 w_0 + \alpha^3 w_1 + \alpha^3 w_1),
\end{aligned}
$$

and

$$V_1(0,1,0|3) = V_1(0,1,0|2) + \alpha^3 w_1.$$

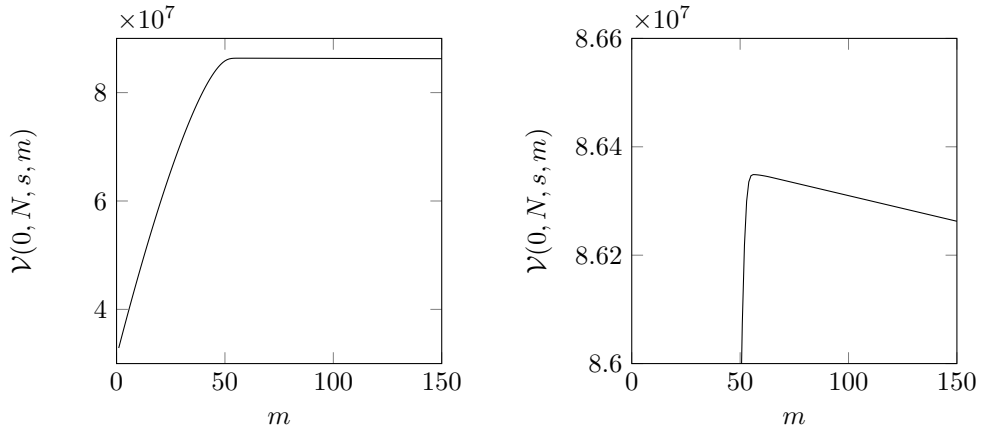
The last expression holds, because the horizon is three periods, and the OEM can use at most two new parts. Now consider the second order difference:

$$
\begin{aligned}
& [V_1(0,1,0|3) - V_1(0,1,0|2)] - [V_1(0,1,0|2) - V_1(0,1,0|1)] \\
& = \alpha^3 w_1 - \left(\alpha^3 w_1 - \lambda_0\lambda_1^2\alpha^3 w_1 + \lambda_0\lambda_1\alpha^2\rho_1 - \lambda_0\lambda_1^2\alpha^2 z + \lambda_0\lambda_1^2\alpha^3\hat{w}_1\right) \\
& = \lambda_0\lambda_1\alpha^2\left[\lambda_1\alpha(w_1 - \hat{w}_1) + \lambda_1 z - \rho_1\right],
\end{aligned}
$$

and let $\lambda_0 = \lambda_1 = 0.03$, $\mu = 0.001$, $\alpha = 0.9$, $w_1 = 1.1$, $w_0 = 0.01$, $z = 1$, $\rho_1 = 0.05$, and $\rho_0 = 0.04$. This yields $[V_1(0,1,0|3) - V_1(0,1,0|2)] - [V_1(0,1,0|2) - V_1(0,1,0|1)] > 0$ and thus no concavity. ◇

However, for practical instances we do observe concavely increasing behavior of $V_1(0,N,s|m)$ and thus also the concavity of $\mathcal{V}(0,N,s,m)$, see Figure 5.1. We have tested the concavity of $\mathcal{V}(0,N,s,m)$ in all our numerical experiments for all $m \in \{1,\ldots,150\}$, where 150 is thrice the number of parts that are installed in the field and therefore we consider it as a reasonable upper bound. Furthermore, we do not consider $m = 0$ as this implies that the OEM prefers Stay Put (not investing in new parts). All our tested instances showed concavity of the value function. Furthermore, testing larger values of m becomes computationally prohibitive. The result in Figure 5.1 is based on the base instance from our numerical experiments in Section 5.6. Moreover, we see that the optimal production quantity m^* is far smaller than our enumerated upper bound of 150. We observe this behavior for all numerical instances that we tested. Moreover, we note that the results from Figure 5.1 show that $\mathcal{V}(0,N,s,m)$ decreases linearly in m, but the profit does not reduce to zero (or even becomes negative). This occurs because we consider values for m up to 150 and the production costs are \$1000; i.e., we approach zero if m is in the order of magnitude of 85,000 new parts.

We believe that the observed concavity is a consequence of the fact that the horizon is long for practical instances, and that the new parts have a higher reliability and



(a) The value function *(b)* The value function (zoomed)

Figure 5.1: Concavity of $\mathcal{V}(0, N, s, m)$ for the base instance from Section 5.6

generate more revenue per time unit. Therefore, there exists a benefit of using new parts over old ones in most practical instances. Under this premises of preferring new parts over old ones, if there are more new parts (higher m), more revenue is generated and therefore $V_1(0, N, s|m)$ increases. The marginal revenue increase of $V_1(0, N, s|m)$ resulting from adding one extra new part reduces: the probability of needing this extra new part decreases. Furthermore, if $m \to \infty$ the OEM salvages all s old parts from stock, because he only installs new parts (and he can always do this as $m \to \infty$). Thus, the OEM earns finite discounted revenue in the initial state $(0, N, s)$, i.e., $\lim_{m\to\infty} V_1(0, N, s|m) < \infty$. This explains why $V_1(0, N, s|m)$ is concavely increasing in m for practical instances, and why $\mathcal{V}(0, N, s, m)$ is concave in m (by its definition), see Figure 5.1. Moreover, we have that $\lim_{m\to\infty} \mathcal{V}(0, N, s, m) = \lim_{m\to\infty} V_1(0, N, s|m) - \lim_{m\to\infty} m\tau = -\infty$.

5.5. Phased Invest

For Phased Invest, we consider two production orders with production quantities m_1 and m_2. The second production order arrives at the start of period L before any event or action has taken place. Therefore, the OEM spreads the production of new parts over two orders, which enables him to postpone the production of some new parts, thereby reducing the costs as there exists discounting. We let $\boldsymbol{m}$ be the vector containing all production quantities (in our case m_1 and m_2). We assume that the OEM pays the same production cost τ for each new part in the second order upon the arrival of the order. This assumption can easily be relaxed such that the unit production costs for each order are different. We illustrate the concept of Phased

Invest with two production orders in Figure 5.2.

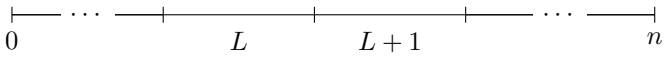


Figure 5.2: An illustration with two production orders

We formulate Phased Invest as a finite horizon discounted DTMDP, where the action space and the one step transition probabilities change once more new parts become available; i.e., the action space and the one step transition probabilities change after the arrival of the m_2 new parts in period L. The state space also changes as new parts arrive, i.e., the state space is augmented once new parts arrive. However, if we consider the general state space $\mathcal{S}_{m_1+m_2} = \{(X_1, N_0, s_0) : 0 \leq X_1 \leq m_1+m_2, 0 \leq N_0 \leq N, 0 \leq s_0 \leq s\}$ for all periods $1, \ldots, n$, we do not need to model the change in the state space, because we cannot reach certain states of the state space in periods $1, \ldots, L-1$ (due to the definition of the action space and one step transition probabilities). For brevity, we write $m_t = m_1 + m_2$ and thus $\mathcal{S}_{m_1+m_2} = \mathcal{S}_{m_t}$. Furthermore, we note that a state $(i, j, k) \in \mathcal{S}_{m_t}$ does not change if more new parts arrive, because the number of new parts does not describe a state.

Let $\mathscr{A}_{m(x)}(\mathfrak{s})$ be the action space for period x, given that we are in state $\mathfrak{s} \in \mathcal{S}_{m_t}$ and where we have $m(x)$ new parts with $m(x) = m_1$ if $x < L$ and $m(x) = m_1 + m_2$ otherwise. The definition of $\mathscr{A}_{m(x)}(\mathfrak{s})$ is given in Eq. (5.3). Finally, we use the definition of the one step transition probability from Instant Invest. The one step transition probability from state $\mathfrak{s} \in \mathcal{S}_{m_t}$ to state $\mathfrak{s}' \in \mathcal{S}_{m_t}$ under action $a \in \mathscr{A}_{m(x)}(\mathfrak{s})$ is given by $p_{m(x)}(\mathfrak{s}'|\mathfrak{s}, a)$.

The OEM is interested in determining the expected maximum discounted profit earned in period 1 in the initial state $(0, N, s)$, given the vector $\boldsymbol{m}$ and given L. Therefore, we let $W_x(\mathfrak{s}|\boldsymbol{m}, L)$ be the maximum expected discounted profit earned over periods $x, x+1, \ldots, n$, given vector $\boldsymbol{m}$ and L, and given state $\mathfrak{s} \in \mathcal{S}_{m_t}$. We have the following optimality recursion

$$W_x(\mathfrak{s}|\boldsymbol{m}, L) = \max_{a \in \mathscr{A}_{m(x)}(\mathfrak{s})} \left\{ \sigma_{m(x)}(\mathfrak{s}, a) + \alpha \sum_{\mathfrak{s}' \in \mathcal{S}_{m_t}} p_{m(x)}(\mathfrak{s}'|\mathfrak{s}, a) W_{x+1}(\mathfrak{s}'|\boldsymbol{m}, L) \right\} - \begin{cases} \tau m_1 & \text{if } x = 1 \\ \tau m_2 \alpha^{L-1} & \text{if } x = L \\ 0 & \text{otherwise.} \end{cases}$$

The optimal decision rule $\eta_x(\boldsymbol{m}, L)$ (for given vector $\boldsymbol{m}$ and L) is determined by

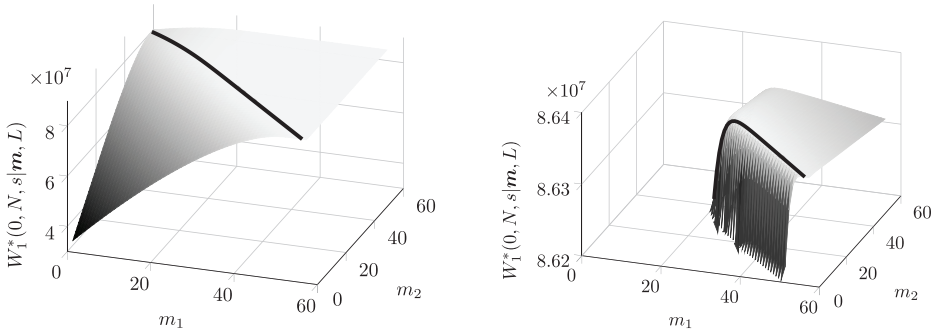
$$\eta_x(\mathfrak{s}|\boldsymbol{m}, L) = \operatorname*{argmax}_{a \in \mathscr{A}_{m(x)}(\mathfrak{s})} \left\{ \sigma_{m(x)}(\mathfrak{s}, a) + \alpha \sum_{\mathfrak{s}' \in \mathcal{S}_{m_t}} p_{m(x)}(\mathfrak{s}'|\mathfrak{s}, a) W_{x+1}(\mathfrak{s}'|\boldsymbol{m}, L) \right\}. \tag{5.4}$$

The terminal vector is given – similar to Instant Invest – by $W_{n+1}(\mathfrak{s}|\boldsymbol{m}, L) = (j+k)w_0 + i\hat{w}_1 + w_1(m_1 + m_2 - i)$, because there are $m_1 + m_2$ working new parts at $n+1$. We solve Phased Invest by enumerating all $W_x(\mathfrak{s}|\boldsymbol{m}, L)$ for each period $x \in \{1, 2, \ldots, n\}$. The objective is to determine the expected discounted profit $W_1(0, N, s|\boldsymbol{m}, L)$ in period 1 for the initial state $(0, N, s)$.

We also shed some light on the optimal production quantities m_1^* and m_2^*, and the optimal arrival time of the second production order L^*. First, we consider the optimization of m_1^* and m_2^* for a given value of L. For given values of m_1, m_2, and L, we use backward induction to determine the optimal decision rule $\eta_x(\boldsymbol{m}, L)$. For the periods $L, L+1, \ldots, n$, $m_1 + m_2$ new parts are available. The production quantity m_1 determines the initial distribution at the start of period L and consequently it influences the optimal production quantity m_2^*. As a result, optimization of m_1^* and m_2^* for a given L is challenging, unless $W_1(0, N, s|\boldsymbol{m}, L)$ is jointly concave in $(m_1, m_2) = \boldsymbol{m}$. However, if $m_2 = 0$, then Phased Invest is equivalent to Instant Invest and Example 5.2 shows that Instant Invest is not concave in m. Consequently Phased Invest cannot be jointly concave. However, it could be the case that joint concavity exists for the practical instances that we consider, but we have to check this numerically. We study the behavior of $W_1(0, N, s|\boldsymbol{m}, L)$ for the base instance of the numerical experiments from Section 5.6, if L is given. First, we determine m^* under Instant Invest and use m^* as a reasonable upper bound for the production quantities m_1 and m_2. Subsequently, we compute the value function $W_1(0, N, s|\boldsymbol{m}, L)$, given L, for all possible values of m_1 and m_2. This takes more than 45 hours in total, and thus it is very time consuming to check $W_1(0, N, s|\boldsymbol{m}, L)$ for all instances, e.g. this would take us approximately 2 months for a given value of L per instance. Hence, we use an approximate approach to determine good or near–optimal production quantities m_1 and m_2 for a given L.

We propose to first determine the optimal production quantity m^* of Instant Invest. Subsequently, we consider various values of L and for each value of L we distribute the production quantity m^* over the two production orders of Phased Invest, i.e., $m_1 + m_2 = m^*$ with $m_1 \geq 1$ and $m_2 \geq 0$. We do not consider $m_1 = 0$ as this means that we are not motivated to move to the new parts (which contradicts Phased Invest). Also, we remark that $m_2 = 0$ means that Phased Invest is equivalent to Instant Invest. Finally, we select the combination L, m_1, m_2 that maximizes the expected profit of Phased Invest such that $m_1 + m_2 = m^*$, and we denote these values by L^*, m_1^*, m_2^*. We have studied the approximation method ($m_1 + m_2 = m^*$) for the base instance from Section 5.6 and $L = 601$. We enumerated all $1 \leq m_1 \leq m^*$ and $0 \leq m_2 \leq m^*$, and plotted the results in Figure 5.3. The solid line is the result for the approximation $m_1 + m_2 = m^*$. The approximation performs well for the base instance with $L = 601$ and supports the fact that we use our proposed heuristic approach to determine m_1^*, m_2^*, and L^* in the remainder.

We conclude this section by a remark that sketches how Phased Invest can be extended to incorporate an arbitrary number of production orders.



(a) The value function *(b)* The value function (zoomed)

Figure 5.3: $W_1(0, N, s|\boldsymbol{m}, L)$ versus $m_1 + m_2 = m^*$ (solid line) for the base instance from Section 5.6 and $L = 601$

Remark 5.2 Let $\kappa \in \mathbb{N}$ be the number of times that the OEM produces new parts. Then, for $1 \leq i \leq \kappa$, let L_i be the period in which the i^{th} order consisting of m_i new parts arrives. The order of m_i new parts arrives at the start of period L_i, before an action or event has occurred. We let $L_1 = 1$ and we define $\boldsymbol{L}$ as the vector containing all L_i. The OEM pays τ for each new part that he produces and the production costs are incurred upon the order's arrival. Then, the value function is given by the recursion

$$W_x(\mathfrak{s}|\boldsymbol{m}, \boldsymbol{L}) = \max_{a \in \mathscr{A}_{m(x)}(\mathfrak{s})} \left\{ \sigma_{m(x)}(\mathfrak{s}, a) + \alpha \sum_{\mathfrak{s}' \in \mathcal{S}_{m_t}} p_{m(x)}(\mathfrak{s}'|\mathfrak{s}, a) W_{x+1}(\mathfrak{s}'|\boldsymbol{m}, \boldsymbol{L}) \right\} - \begin{cases} \tau m_i \alpha^{x-1} & \text{if } x = L_i \\ 0 & \text{otherwise,} \end{cases}$$

with $m(x) = \sum_{j:L_j \leq x} m_j$. The optimal decision rule $\eta_x(\boldsymbol{m}, \boldsymbol{L})$ is determined analogous to Eq. (5.4). Finally, we note that optimization of all m_i^* and all L_i^* can become a highly challenging task, but this is beyond the scope of this chapter.

5.6. Numerical experiments

In this section, we numerically study what strategy is optimal for various parameter settings (instances). We vary the new component's revenue ρ_1, its reliability λ_1, and the unit production cost τ. Subsequently, we briefly discuss the added value of Phased Invest over Instant Invest. Finally, we present results on the number of new parts to produce under Rapid Upgrade, Instant Invest, and Phased Invest.

Our numerical study is based on a limited number of instances, because the

computation time for a single instance is time consuming (approximately 8 hours). This high computation time results from a large state space and that we have to solve all strategies. For each instance, we first compute the expected profit under Stay Put by using the formulation in Appendix 5.A. Secondly, we optimize the number of new parts to produce under Rapid Upgrade and determine the maximum expected profit of this strategy by using Appendix 5.B. Next, we determine the optimal production quantity under Instant Invest m^* and the maximum expected profit, assuming that the value function is concave in m. Finally, we consider Phased Invest and let the second production order arrive one period after a specific fraction of the time horizon (after $5\%, 10\%, 15\%, \ldots, 50\%$ of the horizon), i.e., we consider $L \in \{201, 401, 601, 801, 1001, 1201, 1401, 1601, 1801, 2001\}$. We restrict our considered values for L in order to reduce the required computation time. Then, we enumerate all possible combinations (m_1, m_2) such that $m_1 + m_2 = m^*$ with $m_1 \geq 1$ and $m_2 \geq 0$ for each L. From all these combinations, we are interested in the optimal arrival time of the second production order L^* and the associated optimal production quantities (m_1^*, m_2^*) that maximize the expected profit under Phased Invest.

We consider instances with representative parameter values for parts that are used in capital goods industries. Recall that we assume that at most one event occurs in a period. This assumption has consequences for our numerical study, since the time unit should be small enough such that the assumption is satisfied. Hence, we let a period correspond to one day. Typically, systems are used for a remaining 10–30 years in capital goods industries. We use $n = 4000$ periods as the horizon, which is 10.9 years. Moreover, we consider a relatively small installed base of $N = 50$ parts to obtain reasonable computation times. In addition, the repair costs do not only capture the costs of repairing the failed part, but also incorporate costs due to downtime of a system. For instance, if a failure occurs, a service engineer has to visit the failed system and it takes time to repair the system (say one day). We capture all of these costs in the repair costs z. For Rapid Upgrade, we also have to specify the cost of preventive upgrading d. Typically, a preventive replacement is cheaper than a replacement upon failure, and thus we consider $d < z$. All relevant parameter values of the base instance are given in Table 5.2. Furthermore, we note that $\hat{w}_0$ and $\hat{w}_1$ can be determined via $\hat{w}_0 = w_0 \frac{\alpha\mu}{1-\alpha(1-\mu)}$ and $\hat{w}_1 = w_1 \frac{\alpha\mu}{1-\alpha(1-\mu)}$.

λ_0	λ_1	ρ_0 (\$/day)	ρ_1 (\$/day)	w_0 (\$)	w_1 (\$)	μ	τ (\$)
0.0014	0.0013	1000	1001	200	400	0.015	1000

N	s (parts)	α	z (\$)	d (\$)	n (days)
50	10	0.9995	1200	1100	4000

Table 5.2: Parameter settings of base instance

First, we study which strategy is optimal when varying the performance of the new

component. Therefore, we study the optimal strategy when changing ρ_1 and λ_1. Furthermore, we also look at the optimal strategy when the unit production cost τ changes. We vary the parameters ρ_1, λ_1, and τ unilaterally, and we consider 10 levels other than the base level for each of the parameters. Each of the levels are given in Table 5.3.

λ_1	ρ_1	τ
0.0004	1000.00	500
0.0005	1000.10	600
0.0006	1000.25	700
0.0007	1000.50	800
0.0008	1001.50	900
0.0009	1002.50	1100
0.0010	1004.00	1200
0.0011	1006.00	1300
0.0012	1008.50	1400
0.0014	1011.00	1500

Table 5.3: Levels of variables

The results for the parameter perturbations are given in Figure 5.4, Figure 5.5, and Figure 5.6, respectively. The vertical (y) axis on the right of all figures denotes the expected profit difference with the lowest observed expected profit. We see that our base instance is such that Phased Invest is optimal, since $\lambda_1 = 0.0013$, $\rho_1 = 1001$, and $\tau = 1000$ (see Figures 5.4–5.6). Moreover, we should be cautious when drawing conclusions for the optimal strategy based on the numerical results, because we unilaterally vary the variables and our base instance is such that Phased Invest is optimal.

The results from Figure 5.4 show that Rapid Upgrade becomes attractive when the new component becomes substantially more reliable than the old one (e.g. $\lambda_1 = 0.0004$). That is, if the new parts are much more reliable than the old parts, pursuing Rapid Upgrade is not a bad strategy. However, it is not optimal, because Phased Invest yields higher expected profit (although the expected profit difference with Rapid Upgrade is small). Furthermore, if the difference between ρ_1 and ρ_0 and the difference between λ_1 and λ_0 is small (e.g. $\rho_1 = 1001$, $\rho_0 = 1000$, $\lambda_1 = \lambda_0 = 0.0014$), the OEM can increase the expected profit by \$42,000 if he follows Phased Invest rather than the benchmark strategies Rapid Upgrade or Stay Put. Therefore, the OEM should gradually replace the old parts by the new ones in order to maximize the expected profit. Moreover, the extra profit that is earned (\$42,000) may seem rather small, but we remark that the profit increase is the exclusive result of replacing one component. Systems, typically, consists of multiple components that are replaced by new components. Hence, the profit increase may be substantially higher in practice. Furthermore, we stress that these observations are made under

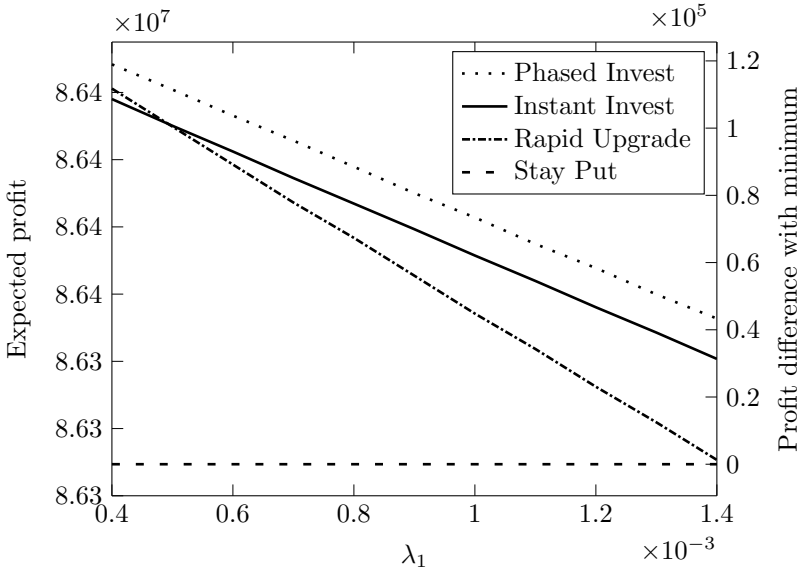


Figure 5.4: The expected profit of all strategies for λ_1 perturbations

the parameter setting that the revenue rate of a new part is slightly higher than the revenue rate of an old part, i.e., $\rho_1 = 1001 > 1000 = \rho_0$. If we were to consider larger or smaller revenue rate differences between an old and new component, we obtain different insights, in which Phased Invest need not be optimal for the considered values of λ_1. For instance, if $\rho_1 = \rho_0$, then there exist $\lambda_1 \leq \lambda_0$ such that Stay Put is optimal (e.g. $\lambda_1 = 0.0013 \leq 0.0014 = \lambda_0$ and $\rho_1 = \rho_0 = 1000$; see Figure 5.5). Yet, the results from Figure 5.4 show that Phased Invest can be optimal if the difference between both components is not too large.

The optimal strategy strongly depends on the revenue rate difference between an old and new component, as Figure 5.5 illustrates. If the revenue rate difference is small or even negligible, there is little incentive to implement the new parts. Hence, we observe that Stay Put is optimal. However, when a new component generates more revenue than an old component, it is optimal to preventively replace the old parts by new ones (Rapid Upgrade). Figure 5.5 shows that Rapid Upgrade is optimal, even if the new component generates only 0.25% ($\rho_1 = 1002.50$) more revenue per period than the old one. Thus, the optimal strategy is sensitive to the relative revenue increase between a new and old component. Furthermore, there also exist ρ_1 values for which both Stay Put and Rapid Upgrade are suboptimal. If the revenue rate difference between an old and new component is (very) small, Phased Invest and Instant Invest are better than the two benchmark strategies (Stay Put and Rapid Upgrade), and in particular Phased Invest is optimal. That is, Phased Invest can increase the expected profit by as much as roughly \$40,000. Again, we note that these numbers can be conservative as systems

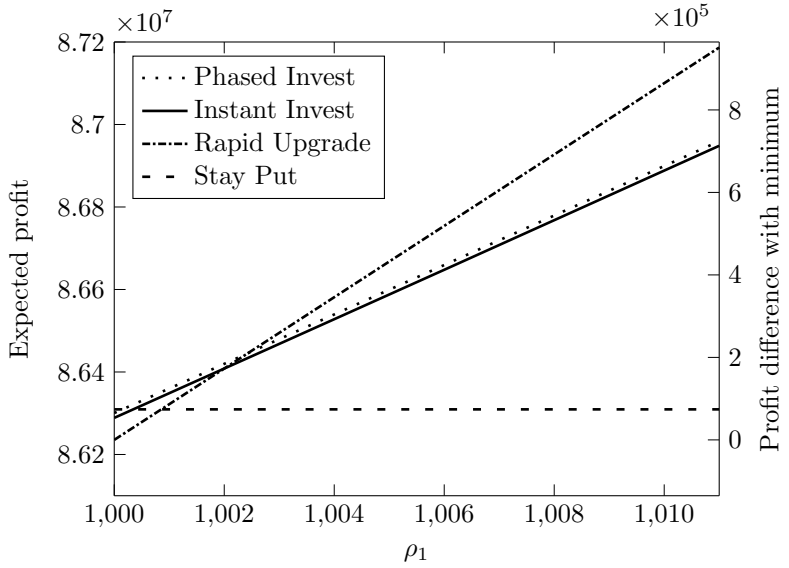


Figure 5.5: The expected profit of all strategies for ρ_1 perturbations

typically consist of multiple components that are replaced by new components. Figure 5.5 illustrates that if the new component generates significantly more revenue than the old one, then the OEM should replace the old parts by new ones as soon as possible, i.e., Rapid Upgrade is a wise strategy to pursue. Stay Put is a good strategy to follow if revenue difference is negligible, indicating old parts should not be replaced by new ones. In cases where the difference between both components is not that extreme, managers should consider Phased Invest and Instant Invest in addition to Stay Put and Rapid Upgrade, because this can increase the expected profit. In other words, Phased Invest is the optimal implementation strategy, implying that the OEM should gradually replace old parts by new ones (upon failure).

If we consider perturbations of the unit production cost τ, we see that Phased Invest is optimal for all our instances. Phased Invest becomes more attractive relative to Rapid Upgrade when the unit production cost increases. This occurs because the value of postponing production increases (more value is postponed as τ increases). For the same reason, the difference between Phased Invest and Instant Invest increases for increasing τ. The difference between Instant Invest and Rapid Upgrade is smaller than the difference between Phased Invest and Rapid Upgrade, because Instant Invest cannot postpone production. Furthermore, we remark that we should be cautious when extrapolating these conclusions, because we have unilaterally varied τ, and the base instance is such that Phased Invest is optimal. Hence, if we would consider a different base instance, Phased Invest need not be optimal for all values of τ; e.g. if $\rho_1 = \rho_0 = 1000$, $\lambda_1 = 0.0013$, $\lambda_0 = 0.0014$ then there exist values for τ for which

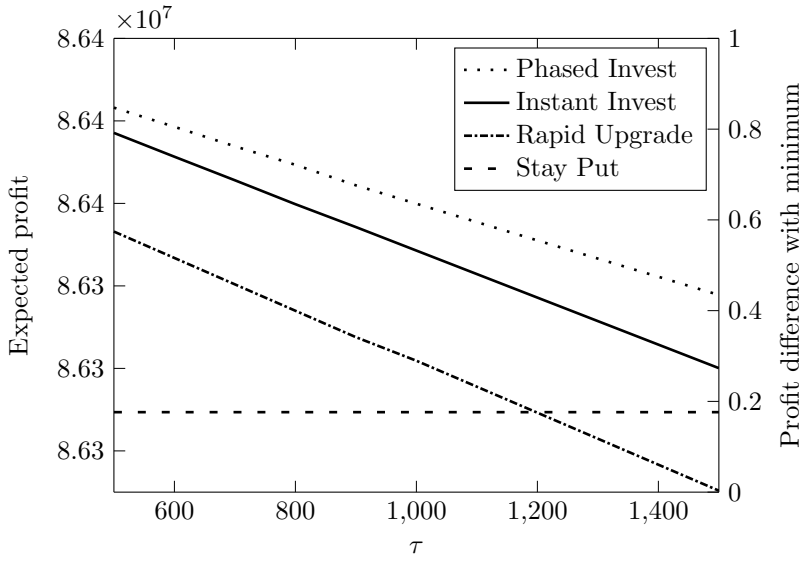


Figure 5.6: The expected profit of all strategies for τ perturbations

Stay Put is optimal, see Figure 5.5. Yet, the results from Figure 5.6 do illustrate how Phased Invest becomes more attractive, compared to Instant Invest and to Rapid Upgrade, once the unit production cost τ increases.

In addition to the foregoing, Figures 5.4–5.6 indicate that the expected profit generated by Instant Invest is rather close to the expected profit generated by Phased Invest. This difference increases (decreases) when the unit production cost τ increases (decreases), because of the value of postponement due to cost discounting. We observe expected profit differences between Phased Invest and Instant Invest in the order of magnitude of \$10,000, which may increase up to roughly \$18,000. However, Phased Invest comes with the drawback that it is difficult to implement, and even more difficult to optimize the production variables L, m_1, and m_2. Hence, the added value of Phased Instant over Instant Invest is rather debatable, because the extra expected profit needs to be traded off against the increased complexity in the formulation of Phased Invest and its optimization complexity.

Finally, we address the optimal production quantity of new parts under Rapid Upgrade $\tilde{m}^*$ and under Instant Invest m^*. Furthermore, we also study the production quantities m_1^* and m_2^*, and the arrival time of the second production order L^* under Phased Invest. The results for each of the instances are given in Table 5.4.

All strategies produce more new parts than the installed base N. Therefore, the OEM keeps new parts on stock under Rapid Upgrade, Instant Invest, and Phased Invest. Furthermore, Rapid Upgrade produces strictly more new parts than Instant Invest and Phased Invest. This is a result from the fact that Rapid Upgrade preventively

		$\tilde{m}^*$	m^*	m_1^*	m_2^*	L^*
λ_1	0.0004	57	53	31	22	601
	0.0005	58	54	31	23	601
	0.0006	59	54	31	23	601
	0.0007	60	55	31	24	601
	0.0008	60	55	31	24	601
	0.0009	61	55	32	23	601
	0.0010	62	56	32	24	601
	0.0011	62	56	33	23	601
	0.0012	63	57	32	25	601
	0.0013	63	57	33	24	601
	0.0014	64	58	32	26	601
ρ_1	1000.00	63	57	32	25	601
	1000.10	63	57	32	25	601
	1000.25	63	57	32	25	601
	1000.50	63	57	33	24	601
	1001.00	63	57	33	24	601
	1001.50	63	57	33	24	601
	1002.50	63	58	33	25	601
	1004.00	63	59	34	25	601
	1006.00	63	60	35	25	601
	1008.50	63	60	36	24	601
	1011.00	63	60	37	23	601
τ	500	64	60	33	27	601
	600	64	59	33	26	601
	700	64	58	33	25	601
	800	64	58	32	26	601
	900	64	57	33	24	601
	1000	63	57	33	24	601
	1100	63	57	33	24	601
	1200	63	57	32	25	601
	1300	63	57	32	25	601
	1400	63	57	32	25	601
	1500	63	57	32	25	601

Table 5.4: Production quantities and the optimal arrival time (Phased Invest)

replaces all old parts by new ones, while Instant Invest and Phased Invest implement the new parts gradually. Hence, the total optimal production quantities of the latter two strategies are lower than $\tilde{m}^*$. Moreover, we observe that the production quantities $\tilde{m}^*$ and m^* are increasing when λ_1 increases, because the demand intensity for new parts increases; thus we need more new parts. Furthermore, the order quantity $\tilde{m}^*$ for Rapid Upgrade does not increase as ρ_1 increases. All old parts are preventively replaced by new ones, and all old parts are immediately salvaged. Hence, there exists no backup if the OEM has no new part available upon a failure, and this is very costly. Therefore, $\tilde{m}^*$ is relatively high for Rapid Upgrade and the changes in $\tilde{m}^*$ are small compared to the lost revenue if the OEM does not have a new part available. Hence, $\tilde{m}^*$ is hardly affected by the changes in ρ_1. For Instant Invest, the optimal quantity m^* increases for increasing revenue ρ_1. If new parts generate more revenue, it is more important to install a new part than an old one. Consequently, the OEM reduces the probability of having no new parts available by increasing m^* under Instant Invest. Finally, the optimal production quantities $\tilde{m}^*$ and m^* decrease as the unit production cost increase, because the production investment increases.

If we study the results for Phased Invest, we note that all optimal arrival times of the second production order L^* are 601 periods (after 15% of the horizon). Also, approximately 60% of the production quantity m^* is produced for the first order and the remaining 40% for the second production order. The results in Table 5.4 further illustrate that the characteristics of the new parts have little influence on L^*, but also have little influence on the distribution of the production quantity m^* over the two production orders m_1^* and m_2^*. Hence, if a manager has determined the optimal values L^*, m_1^* and m_2^*, then these are rather robust for changes in the characteristics of the new parts. So, if the characteristics of new parts are uncertain, the manager could approximate the decision variables of Phased Invest (m_1^*, m_2^*, and L^*) simply by setting $L^* = 601$ and distributing m^* 60%–40% over the two production orders of Phased Invest. This may be particularly useful when implementation strategies are considered during design, or when the failure rates are hard to estimate.

5.7. Conclusion

We studied an OEM that is responsible for a number of systems that are used for a finite remaining lifetime. We focused on critical and repairable parts, and each part occurs once in a system. The OEM keeps a number of parts on stock to respond quickly to failures. The parts that are currently installed or on stock are called *old* parts, and at a certain moment the OEM believes that old parts underperform, i.e., old parts fail too frequently or they do not generate enough revenue. Therefore, the OEM can develop a new component that have better performance (lower failure rates or a higher revenue generation per time unit) compared to the old component. As a result, the OEM has to determine whether and when to implement the new parts. Furthermore, he has to decide how many new parts to produce if he wants to

transition, and how many old parts to salvage from stock. The OEM considers four implementation strategies:

- Stay Put: The OEM does not produce any new parts, repairs the old parts if they fail, and salvages old parts at the end of the horizon.
- Rapid Upgrade: The OEM produces new parts and directly replaces all old parts by new parts. He salvages the old parts immediately (at the start of the horizon). The OEM repairs the new parts once they fail, and he salvages new parts at the end of the horizon.
- Instant Invest: The OEM produces all new parts before the start of the horizon and decides whether to replace a failed part by a new or old one upon each failure. He repairs all new failed parts and old parts are salvaged. Salvaging of old parts occurs during and at the end of the horizon, while salvaging of new parts only occurs at the end of the horizon.
- Phased Invest: The same strategy as Instant Invest, except for the fact that the OEM produces some parts before the horizon and some parts arrive after a number of periods.

We developed a mathematical decision support model that explores all four implementation strategies under given production quantities (for Rapid Upgrade, Instant Invest, and Phased Invest) and given arrival time of the second production order (for Phased Invest). We focused on the formulations for Instant Invest and Phased Invest, because the formulations for Stay Put and Rapid Upgrade are relatively straightforward and these strategies served as benchmark strategies. For Instant Invest and Phased Invest, we presented a finite horizon discounted Markov decision process, and we showed – for both strategies – that new parts (if available) are not necessarily used to replace a failed part, but this does occur in practical instances.

Subsequently, we discussed how to determine the (near) optimal production quantities of new parts (for Rapid Upgrade, Instant Invest, and Phased Invest) and the near–optimal arrival time of the second production order (for Phased Invest). In our numerical experiments we saw that the OEM should not replace the old parts by new ones, if the old and new component are nearly identical in their performance. That is, if both components have near identical failure rates and generate revenue at near identical rates, then Stay Put is a good strategy – and an optimal strategy in many cases. If the difference between both components is large, and in particular if the revenue difference is relatively large, then the OEM should replace the old parts by new ones as soon as possible, i.e., Rapid Upgrade is the optimal strategy. For instances wherein the new component is moderately better than the old one, Phased Invest and Instant Invest are valuable to consider. In such cases, Phased Invest can increase the expected profit notably. Thus, in such cases the OEM should gradually replace the old parts by new ones (upon failure). In addition to studying the optimal implementation

strategy, we also studied the added value of Phased Invest over Instant Invest. We observed that the expected profit generated by Phased Invest is strictly higher than Instant Invest, and this difference can be notable. However, this increase in expected profit comes at a high complexity expense with respect to the formulation of Phased Invest and its optimization.

5.A. Stay Put

Under Stay Put, the OEM produces no new parts and sticks to the old parts. This means that he repairs the old parts during the horizon. We formulate Stay Put as a finite horizon discounted Markov reward chain. The state space $\mathcal{S}'$ is described by the number of old parts in repair, i.e., $\mathcal{S}' = \{1, \ldots, N+s\}$. Furthermore, the one step transition probability from state $\mathfrak{s} \in \mathcal{S}'$ to state $\mathfrak{s}' \in \mathcal{S}'$ is given by

$$p'(\mathfrak{s}'|\mathfrak{s}) = \begin{cases} N\lambda_0 & \text{if } \mathfrak{s}' = \mathfrak{s}+1, \mathfrak{s} \leq s, \\ (N+s-\mathfrak{s})\lambda_0 & \text{if } \mathfrak{s}' = \mathfrak{s}+1, \mathfrak{s} > s, \\ \mathfrak{s}\mu & \text{if } \mathfrak{s}' = \mathfrak{s}-1, \mathfrak{s} \geq 2, \\ 1 - \sum_{\hat{\mathfrak{s}} \in \mathcal{S}' : \hat{\mathfrak{s}} \neq \mathfrak{s}'} p'(\hat{\mathfrak{s}}|\mathfrak{s}) & \text{if } \mathfrak{s}' = \mathfrak{s}, \\ 0 & \text{otherwise.} \end{cases}$$

Subsequently, the revenue $\rho'(\mathfrak{s})$ earned in state $\mathfrak{s} \in \mathcal{S}'$ is derived similar to Instant Invest, and we obtain

$$\rho'(\mathfrak{s}) = \begin{cases} N(\rho_0 - \lambda_0 z) & \text{if } \mathfrak{s} \leq s, \\ (N+s-\mathfrak{s})(\rho_0 - \lambda_0 z) & \text{otherwise.} \end{cases}$$

We use $\rho'(\mathfrak{s})$ along with the one step transition probabilities to determine the total expected profit earned over n periods, with periodic discount factor $0 \leq \alpha \leq 1$. Let $Y_x(\mathfrak{s})$ be the total expected discounted profit earned over periods $x, x+1, \ldots, n$ given that we are in state $\mathfrak{s} \in \mathcal{S}'$. Then, we have the recursion:

$$Y_x(\mathfrak{s}) = \rho'(\mathfrak{s}) + \alpha \sum_{\mathfrak{s}' \in \mathcal{S}'} p(\mathfrak{s}'|\mathfrak{s}) Y_{x+1}(\mathfrak{s}'),$$

with terminal reward $Y_{n+1}(\mathfrak{s}) = \mathfrak{s}\hat{w}_0 + (N+s-\mathfrak{s})w_0$. We assume that the OEM starts in a state wherein the repair queue is empty, i.e., he is interested in $Y_1(0)$. This is analogous to the initial state of Instant Invest and Phased Invest.

5.B. Rapid Upgrade

The formulation for Rapid Upgrade is very similar to the formulation of Stay Put. The difference is that the OEM produces and receives all new parts $\tilde{m}$ prior to the horizon, and the unit production cost of a new part is τ. Next, *all* old parts $N+s$ are preventively replaced by new parts. This means that OEM engineers have to preventively visit and replace N old parts installed in the field. Each preventive replacement of an old part in the field costs d. The costs for the preventive replacement

of the s old parts on stock has no cost. Subsequently, the OEM salvages all $N+s$ old parts at the unit salvage value w_0. Note that we assume here that the OEM starts in an initial state such that no old parts have failed. This is in line with the initial states considered in Stay Put, Instant Invest and Phased Invest. After the old parts have been preventively replaced and the old parts are salvaged, the horizon starts.

We formulate Rapid Upgrade as a finite horizon discounted Markov reward chain. Given $\tilde{m}$, the state space $\tilde{\mathcal{S}}_{\tilde{m}}$ is described by the number of new parts in repair, i.e., $\tilde{\mathcal{S}}_{\tilde{m}} = \{1, \ldots, \tilde{m}\}$. Furthermore, the one step transition probability from state $\mathfrak{s} \in \tilde{\mathcal{S}}_{\tilde{m}}$ to $\mathfrak{s}' \in \tilde{\mathcal{S}}_{\tilde{m}}$ is given by

$$\tilde{p}_{\tilde{m}}(\mathfrak{s}'|\mathfrak{s}) = \begin{cases} N\lambda_0 & \text{if } \mathfrak{s}' = \mathfrak{s}+1, \mathfrak{s} \leq \tilde{m}-N, \\ (\tilde{m}-\mathfrak{s})\lambda_0 & \text{if } \mathfrak{s}' = \mathfrak{s}+1, \mathfrak{s} > \tilde{m}-N, \\ \mathfrak{s}\mu & \text{if } \mathfrak{s}' = \mathfrak{s}-1, \mathfrak{s} \geq 2, \\ 1 - \sum\limits_{\hat{\mathfrak{s}} \in \tilde{\mathcal{S}}_{\tilde{m}}: \hat{\mathfrak{s}} \neq \mathfrak{s}'} \tilde{p}_{\tilde{m}}(\hat{\mathfrak{s}}|\mathfrak{s}) & \text{if } \mathfrak{s}' = \mathfrak{s}, \\ 0 & \text{otherwise.} \end{cases}$$

The revenue earned in state $\mathfrak{s} \in \tilde{\mathcal{S}}$ is similar to Stay Put and given by

$$\tilde{\rho}_{\tilde{m}}(\mathfrak{s}) = \begin{cases} N(\rho_0 - \lambda_0 z) & \text{if } \mathfrak{s} \leq \tilde{m}-N, \\ (\tilde{m}-\mathfrak{s})(\rho_0 - \lambda_0 z) & \text{otherwise.} \end{cases}$$

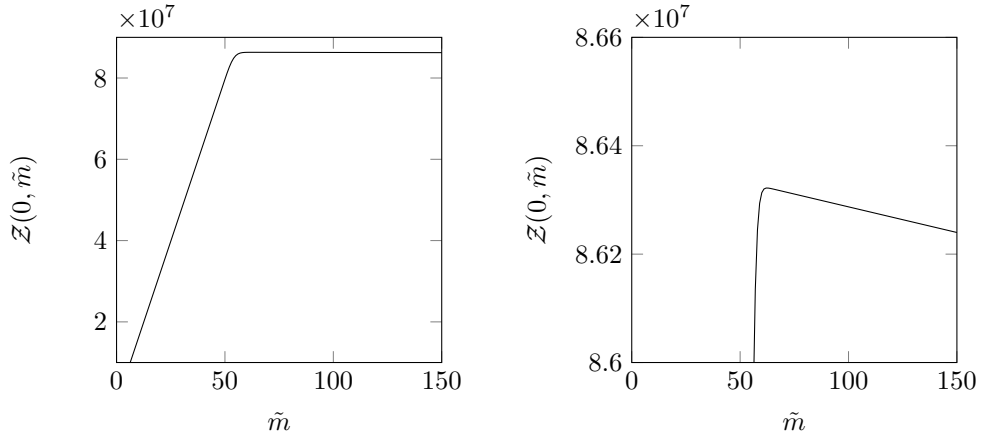
We use $\tilde{\rho}_{\tilde{m}}(\mathfrak{s})$ together with $\tilde{p}_{\tilde{m}}(\mathfrak{s}'|\mathfrak{s})$ in order to determine the total expected discounted profit $Z_x(\mathfrak{s}|\tilde{m})$ earned over periods $x, x+1, \ldots, n$ given state $\mathfrak{s} \in \tilde{\mathcal{S}}$, with periodic discount factor $0 \leq \alpha \leq 1$. Then, we have for the following for the recursion.

$$Z_x(\mathfrak{s}|\tilde{m}) = \tilde{\rho}_{\tilde{m}}(\mathfrak{s}) + \alpha \sum_{\mathfrak{s}' \in \tilde{\mathcal{S}}} \tilde{p}_{\tilde{m}}(\mathfrak{s}'|\mathfrak{s}) Z_{x+1}(\mathfrak{s}'|\tilde{m}),$$

with terminal reward $Z_{n+1}(\mathfrak{s}|\tilde{m}) = \mathfrak{s}\hat{w}_1 + (\tilde{m}-\mathfrak{s})w_1$. Furthermore, the OEM is interested in the expected profit earned given that he start in the initial state where there are no new parts in repair, i.e., $Z_1(0|\tilde{m})$. Finally, the OEM has to consider the costs for producing $\tilde{m}$ new parts, the costs for preventively replacing the old parts by new parts, and the salvage value earned from salvaging $N+s$ old parts. Hence, the total expected profit is $\mathcal{Z}(0, \tilde{m}) = Z_1(0|\tilde{m}) - \tau\tilde{m} - dN + (N+s)w_0$.

The OEM is also interested in the optimization of the production quantity of new parts $\tilde{m}^*$ under Rapid Upgrade. We can bound the expected profit function analogous to Proposition 5.1 and obtain a bound similar to $\overline{m}$. However, such a procedure would result in a loose bound, which offers little computational benefit. Therefore, we explore $Z_1(0|\tilde{m})$. New parts generate non–negative profit $\rho_{\tilde{m}}(\mathfrak{s}) \geq 0$. Thus, increasing the number of new parts, increases $Z_1(0|\tilde{m})$. The concave behavior of $Z_1(0|\tilde{m})$ follows because the marginal profit increase due to one extra new part reduces

(the probability of needing this extra new part reduces if $\tilde{m}$ increases). Hence, we conjecture that $Z_1(0|\tilde{m})$ is concavely increasing in $\tilde{m}$ and thus $\mathcal{Z}(0,\tilde{m})$ is concave in $\tilde{m}$. We have numerically tested this conjecture for $\mathcal{Z}(0,\tilde{m})$ for all instances from Section 5.6, based on $\tilde{m} \in \{1,\ldots,150\}$. For each instance, we observed concavity of $\mathcal{Z}(0,\tilde{m})$ with respect to $\tilde{m}$, and Figure 5.7 illustrates this for the base instance from Section 5.6.



(a) The value function *(b)* The value function (zoomed)

Figure 5.7: Concavity of $\mathcal{Z}(0,\tilde{m})$ for the base instance from Section 5.6

We remark that the results in Figure 5.7 indicate that $\mathcal{Z}(0,\tilde{m})$ decreases linearly in m, but the profit does not reduce to zero (or becomes negative in Figure 5.7), because we only consider values for $\tilde{m}$ up to m and have unit production cost of \$1000; i.e., we approach zero if $\tilde{m}$ is in the order of magnitude of 85,000 parts.

6

Conclusions

In this thesis, we studied decisions that determine costs incurred later in the life cycle. We did so by considering three different research problems (as discussed in Chapter 1). In Chapter 2, we focused on a design problem and we investigated the optimization of commonality and reliability decisions under the presence and absence of service part considerations. In Chapters 3 and 4, we also studied a decision problem that occurs during the design of a system: designing Line Replaceable Units (LRUs). We were interested in determining the optimal design of LRUs such that the downtime and maintenance costs are minimized. Finally, we investigated how to implement system modifications by studying a number of implementation strategies under the presence of service part stocks, in Chapter 5.

In Section 6.1, we discuss the main findings of this thesis, and we conclude this chapter by presenting some potential directions for future research in Section 6.2.

6.1. Main results

In line with the structure of Chapter 1, we discuss the results of each research problem separately. Hence, we summarize the results from Chapter 2 in Section 6.1.1; the results of Chapters 3 and 4 are discussed in Section 6.1.2; and the results from Chapter 5 are summarized in Section 6.1.3.

6.1.1 Service part effects in commonality and reliability decisions

In Chapter 2, we investigated the effects of considering service parts for optimal reliability and commonality decisions, and its consequences on the life cycle costs (see the objective in Section 1.3 of Chapter 1). We took the perspective of an OEM that has closed a service contract with its customers. Furthermore, we focused our attention on two design decisions: whether to use common or dedicated components, and the reliability level for each component (in terms of the mean time between failures). We studied an approach that neglects service parts in design decisions (called non–anticipating), and another approach that considers service parts for design decisions (called anticipating).

For each approach, we presented a model using common components and a model that uses dedicated components. The optimization of the models for the non–anticipating approach are rather straightforward, but the optimization for the models of the anticipating approach are more intricate. Hence, we studied approximate models for the anticipating approach that are asymptotically equivalent as the cost of system downtime tends to infinity.

The models that we developed enabled us to investigate the effect of considering service parts for the optimal reliability decision, the effect of considering service parts for the commonality decision, and the effect of considering service parts for the relevant life cycle costs. We found that if the OEM considers service parts for the reliability decision, he designs more reliable components. The difference in the optimal reliability levels between the non–anticipating (not considering service parts) and anticipating (considering service parts) approach can be as large as 27%, and is on average roughly 10%. Such differences are very substantial and can have large effects on operational costs and after–sales performance. Therefore, the OEM's design engineers should be motivated to consider service parts for the reliability decision. In addition to the results on the optimal reliability levels, we also studied how the commonality decision differs when we consider service parts. We found that commonality is more attractive when service parts are considered for the design decisions. Furthermore, we numerically illustrated that different commonality decisions are made even when the unit cost of the common component increases by as much as 9.59%. So, service parts should be considered if a good commonality decision has to be made: the OEM can use a significantly more expensive common component and still obtain lower life cycle costs under commonality. Finally, our decision support models enabled us to explore how much an OEM can reduce the life cycle costs when considering service parts in commonality and reliability decisions. Our numerical experiments illustrated that the life cycle costs can sometimes be reduced significantly, because neglecting service parts from the commonality and reliability decisions may result in a life cycle cost increase of as much as 10%. Hence, if the OEM wants to improve his profitability, he should encourage his design engineers to consider service parts in the design decisions.

6.1.2 Line Replaceable Units

In Chapters 3 and 4, we studied how to optimally design Line Replaceable Units (LRUs) such that the relevant usage costs are minimized (see the objective in Section 1.3 of Chapter 1). We start with the main results from Chapter 3, after which we discuss the results from Chapter 4. In Chapter 3, we first described how to represent a design of a system for maintenance applications. We proposed to model the parts in a system and their connections as a weighted undirected graph. We also considered a directed acyclic graph that enabled us to incorporate a disassembly sequence in our model. We indicated that this system representation could have the potential to (visually) enhance internal communication at the OEM; for instance, between the design department and the operations department. We used this system representation in Chapter 3 to derive a mathematical decision support model called LRU Design. This model optimizes the design of LRUs such that the total costs of replacing LRUs and purchasing (or repairing) LRUs are minimized. Moreover, we assumed that a part belongs to exactly one LRU, and a LRU is a collection of parts that is replaced when one of the parts in the LRU fails.

We formulated LRU Design as a binary linear program as well as a set partitioning formulation. For the latter formulation we proved that it can be solved by pure pricing algorithms rather than by branch–and–price algorithms, and result in an optimal integer solution. Subsequently, we numerically compared the computation times of both formulations to conclude the set partitioning formulation is applicable for real–life instances, while the binary linear programming formulation is not. This is a direct consequence of the fact that pure pricing algorithms can be used to solve the set partitioning formulation. This makes set partitioning formulation of LRU Design particularly useful as a feedback mechanism for the OEM's design department. The engineers can quickly assess various design alternatives (in terms of the connection graph and precedence graph) and their effects on the optimal LRU design and the corresponding (after–sales) costs. Numerically, we observed that LRUs contain more parts in an optimal solution if the per time unit costs for downtime increase. Moreover, our decision support model LRU Design numerically showed that larger LRUs are optimal if the downtime cost per time unit increase. Also we found that the design of a system has a major impact on the costs that are incurred later in the life cycle. That is, if parts have many connections to other parts or if there exist complex and many disassembly sequences, then the costs in the usage phase (later in the life cycle) substantially increase. Hence, managers should incentivize their designers to reduce the number of connections between parts and to avoid intense disassembly sequences.

In Chapter 4 we used the same representation of a system's design, but we allowed parts to be included in more than one LRU. This means that a LRU, containing a certain part, can be replaced even when the failure of this certain part does not trigger replacement of the LRU. For instance, a motherboard and graphics card are replaced together if the motherboard fails, but if the graphics card fails only the graphics card is replaced. This resulted in a different conceptualization of a LRU, and we used this

new concept of a LRU to derive a model that optimizes the design of LRUs such that the total costs of replacing and purchasing (or repairing) LRUs are minimized. This model was called C–LRU Design.

We proved that C–LRU Design is decomposable in the parts that constitute the system. Hence, we formulated a binary linear program for each of the parts in the system. Subsequently, we solved each program separately to obtain the optimal design of LRUs. We numerically found similar results to Chapter 3, concluding that LRUs contain more parts in an optimal solution if the downtime costs per time unit increase. Furthermore, we saw that a system's design has a large influence on the costs incurred later in the life cycle.

Finally, we compared the computation times, the costs, and the optimal number of LRUs used between LRU Design and C–LRU Design. We observed that computation times increase immensely for LRU Design, by as much as 4555%. The average cost increase when using LRU Design is rather low (approximately 1%), but it can increase up to 26%. These findings make C–LRU Design attractive from a practical perspective, because it can be used as an efficient feedback mechanism to the OEM's design engineers. However, if the number of used LRUs in an optimal solution is essential, both LRU Design and C–LRU Design should be considered because the number of used LRUs can be larger or smaller for C–LRU Design.

6.1.3 Implementation of system modifications

In our final Chapter 5 we shifted our attention from problems that occur in the design phase to a problem present in a life cycle's usage phase. We studied an OEM that closes service contracts with his customers, and we assumed that all service contracts are identical. As a consequence, he is responsible for a number of systems that are installed in the field for a remaining finite time horizon and the OEM is rewarded (penalized) for better (worse) performance. We focused on critical and repairable parts, each of which occurs once in a system. The OEM keeps a number of parts on stock in order to respond quickly to failures. The parts that are operating in the field or are on stock are called *old* parts. At a certain time point the OEM believes that the parts underperform in terms of too high failure rates and/or too little revenue generation. The OEM can design a new component with better performance, and as a result he has to determine whether it is profitable to replace the old parts by new ones. If so, the OEM has to decide when to replace the old parts by new ones. He considers four implementation strategies that dictate whether the OEM should upgrade to the new parts and how fast he should do this, how many new parts to produce, and how many old parts from stock to salvage. Note that these implementation strategies reflect the research objective in Section 1.3 of Chapter 1. The considered implementation strategies are:

- Stay Put: The OEM does not produce any new parts, repairs the old parts if they fail, and salvages old parts at the end of the horizon.

- Rapid Upgrade: The OEM produces new parts and directly replaces all old parts by new parts. He salvages the old parts immediately (at the start of horizon). The OEM repairs the new parts once they fail, and he salvages new parts after the horizon.

- Instant Invest: The OEM produces all new parts before the start of the horizon and decides whether to replace a failed part by a new or old one upon each failure. He repairs all new failed parts and old parts are salvaged. Salvaging of old parts occurs during and at the end of the horizon, while salvaging of new parts only occurs at the end of the horizon.

- Phased Invest: The same strategy as Instant Invest, except for the fact that the OEM produces some parts before the horizon and some parts arrive after a number of periods.

We presented model formulations for each of the implementation strategies under given production quantities (for Rapid Upgrade, Instant Invest, and Phased Invest) and given the arrival time of the second production order (for Phased Invest). These formulations are used to compute the expected generated profit per implementation strategy. Subsequently, we discussed how to determine the optimal production quantities of new parts (for Rapid Upgrade, Instant Invest, and Phased Invest) and the optimal arrival time of the second production order (for Phased Invest).

We numerically illustrated that each of the implementation strategies has its merits. Stay Put is attractive, and optimal in some instances, if the old and new components are nearly identical with respect to the failure rate and the revenue generation, i.e., the OEM should not replace old parts by new ones. Rapid Upgrade, on the other hand, is optimal when the difference between old and new components is significant. Then, the OEM should replace the old parts by new ones as soon as possible. Instant Invest and Phased Invest are valuable when the new component is moderately better than old one, in particular when the revenue generation differs moderately. In these instances, Phased Invest yields strictly higher expected profit than Instant Invest, and the expected generated profit of Phased Invest can be significantly higher than the profit of Rapid Upgrade or Stay Put. The expected profit difference between Instant Invest and Phased Invest is smaller but not necessary negligible. Hence, this should be weighted against the large complexity increase resulting from Phased Invest's formulation. Thus, if the components' performance differ moderately, the OEM should gradually replace old parts by new ones (upon failure).

6.2. Future research

In this thesis we have studied problems in specific and delineated settings. In this last part of the thesis, we step out of this delineation and discuss some of the future directions that research may take. We discuss these directions per problem (analogous to Section 6.1).

In Chapter 2 we used a specific function for the unit production cost (linearly proportional to the relative unit cost of a component). Further research could consider other unit production cost functions. Convexity results may still be attainable for some functions – e.g. if the relative unit cost of a component is additive rather than multiplicative – but in general this need not be the case. Furthermore, the analytical results for the switching curve may no longer be established if the unit production cost function is altered. A second road that future research may take is to model a service part network instead of a single stockpoint. For instance, one could model a multi echelon service part network, potentially with emergency shipments (see for instance van Houtum and Kranenburg (2015)). This could make commonality even more attractive, because the optimal service part stock levels will likely be low and thus pooling could result in larger life cycle costs savings.

The results from Chapter 2 indicate that considering service part stocking is crucial for making the right design decisions. This observation can have implications for further research directions on other design problems, such as the design of Line Replaceable Units. Hence, we suggest researchers to consider service part aspects in future LRU design problems. However, if service part stocking is added to the problems LRU Design and C–LRU Design, Theorem 3.1 from Chapter 3 may no longer hold and neither may the decomposition result in Chapter 4. In addition, it is not obvious how one should embed service part stocking in the models and how this will affect the obtained solutions. The inclusion of service part aspects may introduce non–linearity in the objective functions, which in turn may complicate further analysis substantially. Successive research could explore techniques that are able to cope with this non–linearity of the objective function. A second direction for further research is more theoretical. It could be expected that the LRUs from C–LRU Design in Chapter 4 are nested: if a part v belongs to a replacement set R_Q and to a replacement set $R_{Q'}$, then either $R_Q \subseteq R_{Q'}$ or $R_{Q'} \subseteq R_Q$. We have not explored this theoretical result in the thesis, so further research could study whether such nested behavior exists for the LRUs. Finally, we would like to point out that we assumed that we know exactly and immediately which part has failed once the system fails; i.e., failure diagnostics are instantaneous and perfect and independent of the system design. This assumption could be relaxed in order to come closer to reality. Therefore, we suggest academics to link our problems of designing LRUs to problems studying fault diagnosis, such as the one studied by De Bontridder et al. (2003).

In Chapter 5, we illustrated that a failed part is not always replaced by a new part if available. However, we believe that there may exist certain conditions such that

new parts are preferred (if available) to replace a failed part. Further research could explore whether such conditions exist and what they are. Moreover, further research could study the behavior of the value function for Instant Invest and use insights to improve the optimization of the number of new parts to produce. The same directions could also be taken for Phased Invest, but we think that these steps are more involved and complex compared to Instant Invest. Moreover, research could study the optimal salvage quantity of old parts. This would further reduce the computation times of Instant Invest and Phased Invest, because the action space that needs to be considered for the value functions of both strategies reduces. Finally, the field of condition monitoring may be promising for the development of implementation strategies for new parts. If condition monitoring is able to provide (relatively) accurate information on the failure or maintenance of a part, the OEM can better plan the implementation of a new part. Additionally, this may have an effect on the postponement of production, as he may be able to produce (or purchase) new parts just in time.

Bibliography

R. P. Anstee and M. Farber. Characterizations of totally balanced matrices. *Journal of Algorithms*, 5(2):215–230, 1984.

Y. Asiedu and P. Gu. Product life cycle cost analysis: State of the art review. *International Journal of Production Research*, 36(4):883–908, 1998.

K. R. Baker, M. J. Magazine, and H. L. W. Nuttle. The effect of commonality on safety stock in a simple inventory model. *Management Science*, 32(8):982–988, 1986.

I. Balbaert, A. Sengupta, and M. Sherrington. *Julia: High Performance Programming*. Packt Publishing, 2016.

C. Y. Baldwin and K. B. Clark. *Design rules: The power of modularity*, volume 1. MIT press, 2000.

M. Bijvank, W. T. Huh, G. Janakiraman, and W. Kang. Robustness of order–up–to policies in lost–sales inventory systems. *Operations Research*, 62(5):1040–1047, 2014.

O. Briant and D. Naddef. The optimal diversity management problem. *Operations Research*, 52(4):515–526, 2004.

S. Chand, T. McClurg, and J. Ward. A model for parallel machine replacement with capacity expansion. *European Journal of Operational Research*, 121(3):519–531, 2000.

S. Childress and P. Durango-Cohen. On parallel machine replacement problems with general replacement cost functions and stochastic deterioration. *Naval Research*

Logistics, 52(5):409–419, 2005.

J. Clavareau and P. E. Labeau. Maintenance and replacement policies under technological obsolescence. *Reliability engineering & system safety*, 94(2):370–381, 2009.

CNET News. California power outages suspended – for now, 2001. URL `https://www.cnet.com/news/california-power-outages-suspended-for-now/`.

M. A. Cohen, N. Agrawal, and V. Agrawal. Winning in the aftermarket. *Harvard Business Review*, 84(5):129–138, 2006.

K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. J. Hurkens, J. K. Lenstra, R. Ravi, and L. Stougie. Approximation algorithms for the test cover problem. *Mathematical Programming*, 98(1):477–491, 2003.

T. De Fazio and D. Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, 3(6):640–658, 1987.

Dell Inc. Dell precision 17 7000 series (7710) owner's manual, 2016. URL `http://topics-cdn.dell.com/pdf/precision-m7710-workstation_Owner's%20Manual_en-us.pdf`.

P. Desai, S. Kekre, S. Radhakrishnan, and K. Srinivasan. Product differentiation and commonality in design: Balancing revenue and cost drivers. *Management Science*, 47(1):37–51, 2001.

I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

L. Ellram. A taxonomy of total cost of ownership models. *Journal of business logistics*, 15(1):171, 1994.

R. Fellini, M. Kokkolaras, N. Michelena, P. Papalambros, A. Perez-Duarte, K. Saitou, and P. Fenyes. A sensitivity–based commonality strategy for family products of mild variation, with application to automotive body structures. *Structural and Multidisciplinary Optimization*, 27(1-2):89–96, 2004.

D. R. Fulkerson and R. Hoffman, Oppenheim. On balanced matrices. *Mathematical Programming Study 1*, pages 120–132, 1974.

D. Gross and J. F. Ince. Spares provisioning for repairable items: Cyclic queues in light traffic. *AIIE Transactions*, 10(3):307–314, 1978.

S. Gupta and V. Krishnan. Product family–based assembly sequence design methodology. *IIE transactions*, 30(10):933–945, 1998.

S. Gupta and V. Krishnan. Integrated component and supplier selection for a product family. *Production and Operations Management*, 8(2):163–182, 1999.

J. C. Hartman. The parallel replacement problem with demand and capital budgeting

constraints. *Naval Research Logistics (NRL)*, 47(1):40–56, 2000.

J. C. Hartman and C. H. Tan. Equipment replacement analysis: a literature review and directions for future research. *The Engineering Economist*, 59(2):136–153, 2014.

M. S. Hillier. Component commonality in multiple–period, assemble–to–order systems. *IIE Transactions*, 32(8):755–766, 2000.

A. J. Hoffman, A. W. J. Kolen, and M. Sakarovitch. Totally-balanced and greedy matrices. *SIAM Journal on Algebraic Discrete Methods*, 6(4):721–730, 1985.

C. I. Hsu, H. C. Li, S. M. Liu, and C. C. Chao. Aircraft replacement scheduling: A dynamic programming approach. *Transportation research part E: logistics and transportation review*, 47(1):41–60, 2011.

H. Z. Huang, Z. J. Liu, and D. N. P. Murthy. Optimal reliability, warranty and price for new products. *IIE Transactions*, 39(8):819–827, 2007.

W. T. Huh, G. Janakiraman, J. A. Muckstadt, and P. Rusmevichientong. Asymptotic optimality of order–up–to policies in lost sales inventory systems. *Management Science*, 55(3):404–420, 2009.

P. C. Jones, J. L. Zydiak, and W. J. Hopp. Parallel machine replacement. *Naval Research Logistics*, 38(3):351–365, 1991.

S. H. Kim, M. A. Cohen, and S. Netessine. Performance contracting in after–sales service supply chains. *Management Science*, 53(12):1843–1858, 2007a.

S. H. Kim, M. A. Cohen, and S. Netessine. Reliability or inventory? contracting strategies for after–sales product support. In *Proceedings of 2007 International Conference on Manufacturing & Service Operations Management*, 2007b.

S. H. Kim, M. A. Cohen, and S. Netessine. Reliability or Inventory? An Analysis of Performance–Based Contracts for Product Support Services. In A. Y. Ha and C. S. Tang, editors, *Handbook of Information Exchange in Supply Chain Management*, pages 65–88. Springer International Publishing, 2017.

A. A. Kranenburg and G. J. van Houtum. Effect of commonality on spare parts provisioning costs for capital goods. *International Journal of Production Economics*, 108(1):221–227, 2007.

V. Krishnan and S. Gupta. Appropriateness and impact of platform–based product development. *Management Science*, 47(1):52–68, 2001.

U. D. Kumar, J. Crocker, J. Knezevic, and M. El-Haram. *Reliability, maintenance and logistic support – A life cycle approach.* Springer Science & Business Media, 2012.

E. Labro. The cost effects of component commonality: A literature review through a management–accounting lens. *Manufacturing & Service Operations Management*, 6(4):358–367, 2004.

A. J. D. Lambert. Optimizing disassembly processes subjected to sequence–dependent cost. *Computers & Operations Research*, 34(2):536–551, 2007.

M. Lubin and I. Dunning. Computing in operations research using julia. *INFORMS Journal on Computing*, 27(2):238–248, 2015.

A. Martin-Löf. Optimal control of a continuous–time markov chain with periodic transition probabilities. *Operations Research*, 15(5):872–881, 1967.

G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

S. Mercier. Optimal replacement policy for obsolete components with general failure rates. *Applied Stochastic Models in Business and Industry*, 24(3):221–235, 2008.

S. Mercier and P. E. Labeau. Optimal replacement policy for a series system with obsolescence. *Applied stochastic models in business and industry*, 20(1):73–91, 2004.

A. Mettas. Reliability allocation and optimization for complex systems. In *Reliability and Maintainability Symposium, 2000. Proceedings. Annual*, pages 216–221. IEEE, 2000.

M. H. Meyer and A. P. Lehnerd. *The power of product platforms*. Simon and Schuster, 1997.

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

M. Muffatto and M. Roveda. Developing product platforms: Analysis of the development process. *Technovation*, 20(11):617–630, 2000.

S. K. Nair. Modeling strategic investment decisions under sequential technological change. *Management Science*, 41(2):282–297, 1995.

S. K. Nair and W. J. Hopp. A model for equipment replacement due to technological obsolescence. *European Journal of Operational Research*, 63(2):207–221, 1992.

P. J. Newcomb, B. Bras, and D. W. Rosen. Implications of modularity on product design for the life cycle. *Journal of Mechanical Design*, 120(3):483–491, 1998.

T. K. Nguyen, T. G. Yeung, and B. Castanier. Optimal maintenance and replacement decisions under technological change with consideration of spare parts inventories. *International Journal of Production Economics*, 143(2):472–477, 2013.

R. P. Nicolai and R. Dekker. Optimal maintenance of multi–component systems: A review. In *Complex System Maintenance Handbook*, pages 263–286. Springer London, London, 2008.

G. Norman. Life cycle costing. *Property Management*, 8(4):344–356, 1990.

K. B. Öner, R. Franssen, G. P. Kiesmüller, and G. J. van Houtum. Life cycle costs

measurement of complex systems manufactured by an engineer–to–order company. In R. G. Qui, D. W. Russell, W. G. Sullivan, and M. Ahmad, editors, *The 17th International Conference on Flexible Automation and Intelligent Manufacturing*, pages 569–589. FAIM, Philidelphia, 2007.

K. B. Öner, G. P. Kiesmüller, and G. J. van Houtum. Optimization of component reliability in the design phase of capital goods. *European Journal of Operational Research*, 205(3):615–624, 2010.

K. B. Öner, G. P. Kiesmüller, and G. J. van Houtum. On the upgrading policy after the redesign of a component for reliability improvement. *European Journal of Operational Research*, 244(3):867–880, 2015.

J. E. Parada Puig and R. J. I. Basten. Defining line replaceable units. *European Journal of Operational Research*, 247(1):310–320, 2015.

K. Parent. Avantcom moves on pilot project, 2000. URL `http://www.edn.com/electronics-news/4362659/AvantCom-Moves-on-Pilot-Project`.

D. A. Patterson. A simple way to estimate the cost of downtime. In *LISA '02 Proceedings of the 16th USENIX conference on System administration*, pages 185–188. USENIX Association, 2002.

S. Rajagopalan, M. R. Singh, and T. E. Morton. Capacity expansion and replacement in growing markets with uncertain technological breakthroughs. *Management Science*, 44(1):12–30, 1998.

J. O. Royset, A. Der Kiureghian, and E. Polak. Reliability–based optimal structural design by the decoupling approach. *Reliability Engineering & System Safety*, 73(3): 213–221, 2001.

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

K. Selviaridis and F. Wynstra. Performance-based contracting: a literature review and future research directions. *International Journal of Production Research*, 53 (12):3505–3540, 2015.

D. M. Sharman and A. A. Yassine. Characterizing complex product architectures. *Systems Engineering*, 7(1):35–60, 2004.

C. C. Sherbrooke. *Optimal Inventory Modeling of Systems: Multi–Echelon Techniques*. Kluwer, Dordrecht, 2004.

J. S. Song and Y. Zhao. The value of component commonality in a dynamic inventory system with lead times. *Manufacturing & Service Operations Management*, 11(3): 493–508, 2009.

M. E. Sosa, S. D. Eppinger, and C. M. Rowles. A network approach to define

modularity of components in complex products. *Journal of Mechanical Design*, 129(11):1118–1129, 2007.

D. V. Steward. The design structure system: A method for managing the design of complex systems. *IEEE transactions on Engineering Management*, (3):71–74, 1981.

J. M. Swaminathan and S. R. Tayur. Managing broader product lines through delayed differentiation using vanilla boxes. *Management Science*, 44(12):S161–S172, 1998.

L. Thomas. A survey of maintenance and replacement models for maintainability and reliability of multi–item systems. *Reliability Engineering*, 16(4):297–309, 1986.

U. W. Thonemann and M. L. Brandeau. Optimal commonality in component design. *Operations Research*, 48(1):1–19, 2000.

L. A. M. van Dongen. Maintenance engineering: Instandhouding van verbindingen. Oratie, 2011.

G. J. van Houtum and A. A. Kranenburg. *Spare parts inventory control under system availability constraints*, volume 227. Springer, 2015.

J. A. van Mieghem. Commonality strategies: Value drivers and equivalence with flexible capacity and inventory substitution. *Management Science*, 50(3):419–424, 2004.

T. Zou and S. Mahadevan. Versatile formulation for multiobjective reliability–based design optimization. *Journal of Mechanical Design*, 128(6):1217–1226, 2006.

Summary

Life Cycle Costs Optimization for Capital Goods

The investment decision for systems that require large financial investments – e.g. an airplane or a lithography system – is involved and multidimensional, because customers do not only consider the initial purchase costs. Rather, customers consider operational aspects for such systems, because the operational costs can account for the majority of the total cost of owning systems. These operational costs consist largely of maintenance costs, repair costs, and downtime costs. The costs for usage – e.g. operator and facility costs – are excluded from the operational costs. Since the operational costs are high, customers require that Original Equipment Manufacturers (OEMs) offer a total package. Therefore, OEMs close service contracts with customers, and these contracts make the OEM responsible for a major part of systems' life cycles. The service contracts are such that both parties (OEM and customer) benefit. As a consequence, OEMs are no longer solely focused on offering systems with a low initial purchase price, but are offering solutions in which the life cycle costs of systems are minimized.

The total life cycle costs are a result of various decisions that are taken during various phases of the life cycle, e.g. during design, production or usage. The majority of these decisions does not only influence the immediate costs, but also determines – to a large extent – the costs that are incurred later in the life cycle. In this thesis, we study three problems for which decisions determine the costs incurred later in the life cycle. Each of the studied problems is difficult to solve without the help of advanced mathematical decision support models. Therefore, we develop a mathematical decision support model for each problem that aids the OEM in an attempt to lower (part of) the life cycle costs by looking forward in the life cycle.

Service part effects in commonality and reliability decisions

In Chapter 2, we study a problem in the design phase of a component. In particular, we explore the value of considering service part stocks for commonality and reliability decisions. Moreover, we investigate how much the life cycle costs can be reduced if an OEM considers service parts in the commonality and reliability decisions. The OEM should decide whether to use a common component (one component for all systems) or multiple dedicated components (one component per system). Furthermore, he determines the reliability for each alternative.

We propose two approaches: one in which the OEM considers costs for production and repair, and where he neglects service parts for the commonality and reliability decisions. In the second approach, the OEM considers production and repair costs, as well as holding costs and downtime costs because the OEM includes service parts in his commonality and reliability decisions. For the former approach – in which the OEM does not consider service parts – we optimize only the commonality decision and the reliability levels, which is analytically tractable. Under the latter approach, the optimization models (for common components and dedicated components) that optimize the commonality decision, the reliability levels and the service part inventory levels are analytically intractable. Therefore, we use the observation that downtime of systems is expensive for capital intensive systems. Consequently, we study two approximate models (common and dedicated) and for each of them we prove that it is asymptotically equivalent to the original model (common or dedicated) as the cost for downtime tends to infinity. Using the approximate models, we are able to find and characterize the optimal commonality decision, optimal reliability levels and the service part inventory levels.

We compare the optimal reliability levels under both approaches, and we observe that considering service parts can result in optimal reliability levels (in terms of mean time between failures) that are 27% higher than the ones found under neglecting service parts. In addition, we analytically characterize a switching curve that determines the optimal commonality decision under both approaches. We prove that commonality is selected strictly more when service parts are considered for the reliability and commonality decisions, and numerically we illustrate that we can obtain a different commonality decision even when the unit cost of a common component increases by 9.5%. Finally, we illustrate that considering service parts in the reliability and commonality decisions may lead to much lower relevant life cycle costs, being as much as 10% lower in specific cases.

Line Replaceable Units

In Chapters 3 and 4, we study the design of Line Replaceable Units (LRUs). A LRU is a collection of parts that is replaced entirely when one of the parts in the LRU has to be maintained. The design decision of LRUs is typically made late(r) in the

design phase of a life cycle, where the design of a system is used as input. The design of LRUs has an impact on the operational costs that are incurred later in the life cycle: the design of LRUs determines the downtime and maintenance costs, which are predominant cost factors in the usage phase of a life cycle.Therefore, a good LRU design is essential for lowering the total life cycle costs. The cost effects of a certain LRU design are twofold. On the one hand, LRUs can lower the time an engineer spends on replacing the failed parts. If certain parts are combined together in an LRU it may be easier and faster to replace them, thereby lowering the downtime. On the other hand, failed LRUs are replaced by ready–for–use ones that have been purchased or repaired. When LRUs are small they contain less value than larger LRUs, and thus the cost for purchasing or repairing increases for larger LRUs.

In Chapter 3 we represent a technical system with its existing disassembly characteristics in terms of two graphs. A system consists of multiple parts that are connected to each other, and each part has a failure rate and a purchase cost. Furthermore, breaking and re–establishing a connection between two parts costs time (and money), which we use as edge weights. We represent a system as an undirected graph, where each part corresponds to a vertex with a failure rate and purchase cost, and a connection is an edge with an edge weight. Furthermore, we include the disassembly sequence that exists for maintenance (e.g. remove part A before part B) by using a directed acyclic graph. With the system representation in place, we make an important assumption in Chapter 3 that each part belongs to exactly one LRU. Next, we use both graphs to derive an optimization model called LRU Design, and this model optimizes the design of LRUs such that the total replacement costs and total purchase (or repair) costs are minimized. We present the most natural binary non–linear programming formulation (BNLP) for this problem. Subsequently, we linearize the BNLP to obtain a binary linear programming (BLP) formulation. Furthermore, we present a set partitioning formulation of LRU Design that allows for branch–and–price algorithms. We prove that branching is unnecessary to obtain an optimal integer solution for the set partitioning formulation. Moreover, we compare the computation times of the two formulations (BLP and set partitioning) in our numerical experiments to conclude that the set partitioning formulation is suitable for large instances, while the BLP formulation is not. Finally, we illustrate (in Chapter 3) that higher downtime costs per time unit result in larger LRUs, and that it is crucial to reduce the number of connections and the complexity of the disassembly sequences in order to lower the life cycle costs.

In Chapter 4, we relax the assumption that each part belongs to exactly one LRU. Consider, for instance, a student who owns a city bike. If one of the spokes of the rear wheel breaks, the student decides to only replace the broken spoke. However, if the rim breaks, the student replaces the entire rear wheel (consisting of a rim, all spokes, a wheel hub, and a cassette). Hence, a spoke is replaced upon its own failure *or* upon the failure of the rim. Therefore, we say that a spoke belongs to more than one LRU. This differs from our approach in Chapter 3, and thus we conceptualize a LRU differently in Chapter 4: we describe a LRU by a tuple consisting of a failure

set and a replacement set. All parts in the replacement set are replaced if any of the parts from the failure set fails (this also means that the failure set is a subset of the replacement set). Using this new conceptualization of a LRU, we present an optimization model called C–LRU Design that optimizes the design of LRUs such that the total replacement and purchase (or repair) costs are minimized. We prove that C–LRU Design is separable in the number of parts constituting the system. Using this decomposition result, we present a binary linear program for each part and we numerically show that this formulation is efficient, even for large instances. We find that C–LRU Design behaves the same as LRU Design to changes in the downtime costs per time unit, the number of connections and the complexity of the disassembly sequences. The computation times of the set partitioning formulation of LRU Design increase dramatically compared C–LRU Design, by as much as 4555%. Furthermore, we observe that LRU Design can substantially increase the costs by as much as 27%.

Implementation of system modifications

In Chapter 5, we study an OEM responsible for a number of systems used for a finite horizon, in the order of magnitude of 10–30 years. We focus on critical and repairable parts of a single component, and each part occurs once in a system. The OEM keeps a number of parts on stock to respond quickly to failures. The parts that are currently installed or on stock are called *old* parts. At a certain time, the OEM observes that the old parts underperform, and thus a new component is designed. New parts fail less frequently and/or generate more revenue than old parts. Hence, the OEM has to decide whether and when to replace the old parts by the new parts and what to do with the old parts on stock. Therefore, he considers the following implementation strategies:

- Stay Put: The OEM does not produce any new parts, repairs the old parts if they fail, and salvages old parts at the end of the horizon.
- Rapid Upgrade: The OEM produces new parts and directly replaces all old parts by new parts. He salvages the old parts immediately (at the start of the horizon). The OEM repairs the new parts once they fail, and he salvages new parts at the end of the horizon.
- Instant Invest: The OEM produces all new parts before the start of the horizon and decides whether to replace a failed part by a new or old one upon each failure. He repairs all new failed parts and old parts are salvaged. Salvaging of old parts occurs during and at the end of the horizon, while salvaging of new parts only occurs at the end of the horizon.
- Phased Invest: The same strategy as Instant Invest, except for the fact that the OEM produces some parts before the horizon and some parts arrive after a number of periods.

We present a model that explores all four implementation strategies, and we focus on the formulations of Instant Invest and Phased Invest because Stay Put and Rapid Upgrade are fairly easy formulations and they serve as benchmark strategies. For Instant Invest and Phased Invest, we formulate a finite horizon, discounted, discrete time Markov decision process that maximizes profit. For these two strategies, we show that it is not necessarily optimal to replace a failed part by a new part (if available). However, for practical instances, we do find that new parts (if available) are used to replace a failed one. Furthermore, we discuss how to optimize the production quantities of new parts (for Rapid Upgrade, Instant Invest, and Phased Invest) and the arrival time of the second production order (for Phased Invest).

In our numerical experiments, we illustrate that Stay Put is a good strategy if the old and new component have nearly identical failure rates and generate revenue at nearly identical rates. In this case, the OEM is not motivated enough to implement the new parts. If the difference in the revenue rate between both components is large, then Rapid Upgrade is the optimal strategy to pursue: the OEM should use the new parts as soon as possible. For instances in which the new component is marginally better than the old one, Phased Invest and Instant Invest can increase the expected profit notably. Thus, the OEM should gradually implement the new parts. Finally, we find that Phased Invest generates strictly more profit than Instant Invest and this difference can be notable, but it should be weighted against the increased complexity of Phased Invest's formulation and its optimization.

About the author

Joni Driessen was born in Helden (The Netherlands) on November 23, 1988. He finished his pre–university education at the Willibrord Gymnasium in Deurne (The Netherlands) in 2007. Thereafter, he obtained a BSc in Industrial Engineering and Management Science in 2011 and a MSc in Operations Management and Logistics (cum laude) in 2014, both from Eindhoven University of Technology (The Netherlands).

In 2014, he started his PhD research at the same university under the supervision of prof.dr.ir. Geert–Jan van Houtum and dr.ir. Joachim Arts. During this PhD project, Joni cooperated with a number of companies, in particular with ASML and Dutch Railways. Furthermore, he visited Carnegie Mellon University in Pittsburgh (Pennsylvania, United States) for four months to work with prof.dr. Alan Scheller–Wolf.