An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies

Jussi Leppinen, Antti Punkka, Tommi Ekholm, Ahti Salo

PII: S0305-0483(24)00127-0
DOI: https://doi.org/10.1016/j.omega.2024.103162
Reference: OME 103162

To appear in: Omega

Received date: 26 January 2023
Accepted date: 20 July 2024

Please cite this article as: J. Leppinen, A. Punkka, T. Ekholm et al., An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies. Omega (2024), doi: https://doi.org/10.1016/j.omega.2024.103162.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
Abstract

In most multi-component systems, the cost-efficiency of maintenance policies depends on technical structural dependencies. Motivated by the recognition that these dependencies must be accounted for in the development of optimal maintenance policies, we develop an optimization model to determine cost-efficient maintenance schedules for multi-component systems. Our main contribution is twofold. First, we introduce directed graphs as an expressive tool to represent the economic and structural dependencies of the system, including situations in which the maintenance of a given component may require other components to be disassembled or maintained. Second, we formulate a Markov Decision Process model, which is solved through the modified policy-iteration algorithm to determine the most cost-efficient policy. This policy indicates which maintenance actions consisting of disassembly and component replacement decisions are optimal when mandatory replacements must be made whenever the system fails, or the reliability of the system falls below a predefined reliability threshold. To our knowledge, this is the first model that provides optimal maintenance policies that comply with reliability requirements in the presence of constraints arising from technical structural dependencies. We illustrate the model with a realistic case study on the development of cost-efficient maintenance policies and show that its results compare favorably with heuristic maintenance policies.

Keywords: Maintenance scheduling, multi-component system, maintenance action portfolio, Markov Decision Process, modified policy-iteration
1. Introduction

Technical multi-component systems call for maintenance actions for reliable and safe operation. In general, the components of the system wear out at different rates. Thus, it is not necessarily cost-efficient to maintain all components simultaneously to ensure requisite reliability. On the other hand, the costs of system downtime and maintenance set-up, as well as economic dependencies between the maintenance decisions for different components, imply that it can be cost-efficient to maintain more than one component simultaneously (see, e.g., de Jonge and Scarf, 2020). For example, the schedules for car maintenance – which are based on the driving distance or the time from the previous maintenance instance – suggest that several components should be maintained at the same time, although these components are unlikely to deteriorate at the same pace.

More generally, there is a need to combine preventive maintenance schedules with unscheduled corrective maintenance caused by possible component failures. The scheduling can be further complicated by, for example, (i) reliability requirements (e.g. Nguyen et al., 2015; Shi et al., 2020; Wang et al., 2022), (ii) economic dependencies and (iii) structural dependencies between components (e.g. Olde Keizer et al., 2017a; de Jonge and Scarf, 2020).

Many methods have been developed to account for economic dependencies in the maintenance schedules over a long time horizon (de Jonge and Scarf, 2020). These methods can accommodate maintenance cost and system reliability either as separate objectives (e.g. Nguyen et al., 2015), treat one of these two objectives as a constraint (see Wang, 2002, for a survey), or assign a cost to the risk of failure (e.g. Wildeman et al., 1997; Van Horenbeek and Pintelon, 2013; Vu et al., 2014). Structural dependencies are often considered from the perspectives of how the maintenance of some groups of components can cause a system shutdown (e.g. Van Horenbeek and Pintelon, 2013; Nguyen et al., 2015) or how the total maintenance cost of the group depends on structural dependencies (Geng et al., 2015). Indeed, Nicolai and Dekker (2008) and de Jonge and Scarf (2020) suggest that scheduling models should account for technical structural dependencies such that the maintenance of a component calls for the simultaneous disassembly or maintenance of other components. Disassembly has received much attention recently (e.g. Zhou et al., 2015; Dao and Zuo, 2017; Dinh et al., 2020a,b, 2022, 2024). Specifically, the impacts of technical structural dependencies cannot be fully understood without considering the possibility of simultaneous replacements. For example, it may be advantageous to replace a bicycle chain and its adjoining cassette simultaneously (Nicolai and Dekker, 2008).

Methods for determining maintenance policies have been largely based on heuristic optimization and simulation. While heuristics have proven useful in many instances, a single heuristic provides cost-efficient maintenance schedules only in a specific problem setting. For example, optimal maintenance schedules for
single components cannot usually be applied to multi-component systems due to dependencies between components (Quatrini et al., 2020). Yet, maintaining components simultaneously is likely to be sub-optimal if the decision is based on a predetermined policy based on threshold levels (Olde Keizer et al., 2017a). Furthermore, the choice of a heuristic requires insights into the characteristics of the system. Against this backdrop, de Jonge and Scarf (2020) note that further research is needed to improve these heuristics and, ideally, to determine optimal maintenance scheduling policies.

One way to determine maintenance scheduling policies is to model the system and maintenance decisions as a **Markov Decision Process** (MDP; see Howard, 1960). MDPs have been built to determine optimal maintenance policies for systems consisting of one (e.g., Chan and Asgarpoor, 2006) or two components (e.g., van der Duyn Schouten and Vanneste, 1990; Zhang et al., 2012; Broek et al., 2021). Olde Keizer et al. (2016, 2018) and Xu et al. (2021) consider economical dependencies in building a MDP to solve optimal maintenance schedules for a **multi-component** system in which $k$-out-of-$N$ components must be operative for the system to operate. These schedules are not based on predetermined thresholds. Still, to our knowledge, none of the earlier MDP-based maintenance scheduling models have considered the reliability of the system in conjunction with the technical structural dependencies that arise from the disassembly and replacement requirements.

In this paper, we develop a MDP model for determining cost-efficient maintenance schedules for multi-component systems in the presence of economic and structural dependencies between components. The dependencies are modeled as a directed graph whose nodes represent maintenance actions, arcs depict structural dependencies, and weights attached to arcs represent the costs of corresponding maintenance actions. Thus, this graph makes it possible to specify structural conditions for replacing a component. For example, it may be that component $i$ can be replaced only on condition component $j$ is disassembled. Additionally, the cost of replacing $i$ may depend on whether $j$ is disassembled or replaced. The maintenance actions consist of component (i) replacement decisions and (ii) component disassembly decisions. Both are possible only at predefined maintenance instances (for comparison, see, e.g., Wee and Widyadana, 2013; Baldi et al., 2016). Thus, our model is most accurate in settings in which maintenance services are a scarce, prebooked resource.

We assume that the maintenance actions are perfect in the sense that replacements fully restore the condition of components and that these actions do not impact the failure processes of other components. In addition, we assume that the system operates only when every component operates. As usual, we assume that components deteriorate over time. Thus, older components are more likely to fail. We set a **reliability threshold** to limit system failure risk. At each maintenance instance, preventive maintenance actions (i.e., component replacements) may be needed to ensure that there is a sufficiently high probability
that the system remains operational until the next maintenance instance.

We determine the optimal maintenance scheduling policy with the modified policy-iteration algorithm, which computes the feasible maintenance action portfolios that minimize the total maintenance costs in the long run. The computational performance of this algorithm is enhanced by implementing both the Gauss-Seidel (Puterman, 2014) and Anderson acceleration (Geist and Scherrer, 2018) to speed up the value function updates and thus the convergence of the algorithm. The implementation of the Anderson acceleration expands the value function update schemes considered in Andersen et al. (2022). At each maintenance instance, the resulting maintenance policy accounts for a possible component failure and also all possible future states with associated probabilities, thus keeping the schedule up-to-date. In addition to guiding maintenance decisions at the present maintenance instance, the maintenance policy helps the decision-maker prepare for future maintenance needs by providing information about possible future maintenance action portfolios and the probabilities with which these will be implemented. Furthermore, the model can be used as a benchmark for different maintenance instance strategies, which, for example, different maintenance providers can use (Gouveia et al., 2015).

The remainder of this paper is organized as follows. Section 2 provides a literature review. Section 3 develops the MDP model and presents the algorithm. Section 4 presents an illustrative example, examines the properties of the optimal policy, compares this policy with two simple heuristic policies, and presents computational illustrations. Section 5 discusses alternative uses of the optimization model for decision support and charts avenues for extending the model. Section 6 concludes.

2. Maintenance scheduling in the literature

Maintenance scheduling can offer major performance improvements and cost savings as an essential business function. Recently, models for maintenance scheduling have received increasing attention (see e.g. de Jonge and Scarf, 2020; Quatrini et al., 2020). While different application areas call for different maintenance scheduling models, these models usually accommodate features that are common to many maintenance problems. In this section, we consider models for the maintenance scheduling of multi-component systems based on (i) economic dependencies, (ii) structural dependencies, (iii) reliability thresholds, and (iv) solution methods. Table 1 summarizes models discussed in this section.
Leppinen et al.

Table 1: Models for the optimal maintenance scheduling of multi-component systems with economic and structural dependencies. Economic dependence can be positive (+) or negative (−). Structural dependence can be technical (tech.) and/or related to performance (perf.). The reliability threshold can apply to single components or the whole system. Solution methods are described in Section 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dependence</th>
<th>Reliability</th>
<th>Solution method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vu et al. (2014)</td>
<td>+, −</td>
<td>perf.</td>
<td>rolling horizon, GA</td>
</tr>
<tr>
<td>Do et al. (2015b)</td>
<td>+, −</td>
<td>perf.</td>
<td>rolling horizon, GA</td>
</tr>
<tr>
<td>Van Horenbeek and Pintelon (2013)</td>
<td>+</td>
<td>perf.</td>
<td>dynamic rolling horizon</td>
</tr>
<tr>
<td>Geng et al. (2015)</td>
<td>+, −</td>
<td>tech.</td>
<td>parametric simulation</td>
</tr>
<tr>
<td>Zhou et al. (2015)</td>
<td>+, −</td>
<td>tech.</td>
<td>parametric simulation</td>
</tr>
<tr>
<td>Nguyen et al. (2015)</td>
<td>+, −</td>
<td>perf.</td>
<td>parametric simulation</td>
</tr>
<tr>
<td>Chalabi et al. (2016)</td>
<td>+</td>
<td>perf.</td>
<td>PSO</td>
</tr>
<tr>
<td>Olde Keizer et al. (2016)</td>
<td>+</td>
<td>perf.</td>
<td>dynamic programming</td>
</tr>
<tr>
<td>Dao and Zuo (2017)</td>
<td>+, −</td>
<td>tech.</td>
<td>system</td>
</tr>
<tr>
<td>Olde Keizer et al. (2018)</td>
<td>+</td>
<td>perf.</td>
<td>MDP, VI</td>
</tr>
<tr>
<td>Shi et al. (2020)</td>
<td>+</td>
<td>perf.</td>
<td>system</td>
</tr>
<tr>
<td>Liang and Parlak (2020)</td>
<td>+</td>
<td>perf.</td>
<td>rolling horizon, GA</td>
</tr>
<tr>
<td>Vu et al. (2020)</td>
<td>+, −</td>
<td>perf.</td>
<td>rolling horizon, GA</td>
</tr>
<tr>
<td>Safaci et al. (2020)</td>
<td>perf.</td>
<td>system</td>
<td>copula framework</td>
</tr>
<tr>
<td>de Pater and Mitici (2021)</td>
<td>perf.</td>
<td>system</td>
<td>rolling horizon, ILP</td>
</tr>
<tr>
<td>Xu et al. (2021)</td>
<td>perf.</td>
<td>syst., comp.</td>
<td>PSO, parametric simulation</td>
</tr>
<tr>
<td>Wang et al. (2022)</td>
<td>perf.</td>
<td>tech.</td>
<td>system</td>
</tr>
<tr>
<td>Dinh et al. (2020b)</td>
<td>perf.</td>
<td>component</td>
<td>parametric simulation</td>
</tr>
<tr>
<td>Dinh et al. (2022)</td>
<td>tech.</td>
<td>component</td>
<td>PSO, evaluation with simulation</td>
</tr>
<tr>
<td>Andersen et al. (2022)</td>
<td>perf.</td>
<td>system</td>
<td>MDP, VI, PI, MPI</td>
</tr>
<tr>
<td>Urbani et al. (2023)</td>
<td>perf.</td>
<td>system</td>
<td>rolling horizon, GA</td>
</tr>
<tr>
<td>Zheng et al. (2023)</td>
<td>perf.</td>
<td>system</td>
<td>MDP, VI</td>
</tr>
<tr>
<td>This work</td>
<td>perf.</td>
<td>system</td>
<td>MDP, MPI</td>
</tr>
</tbody>
</table>

Maintenance models can be developed for one-component or multi-component systems. Multi-component systems can have economic, stochastic, structural, and resource dependencies between components that affect the optimal maintenance schedules (Olde Keizer et al., 2017a). In this paper, we focus on economic and structural dependencies and exclude the consideration of stochastic and resource dependencies.

Economic dependence means that the cost of maintaining a group of components simultaneously differs from the cost of maintaining the components separately (de Jonge and Scarf, 2020). Economic dependence is positive (negative) if maintaining the components in the group simultaneously is less (more) expensive than maintaining them separately (Olde Keizer et al., 2017a). Positive economic dependence is usually modeled with a maintenance set-up cost, which can be saved when components are maintained simultaneously rather than separately (Wildeman et al., 1997).

Structural dependence means that some operational components must be replaced or at least dis-
assembled before failed components can be maintained (Nicolai and Dekker, 2008). Olde Keizer et al. (2017a) distinguish between performance dependence and technical dependence. Specifically, performance dependence means that different configurations of components can affect the reliability and availability of the system. For example, in a series system, all components must operate for the system to operate, whereas, in a parallel system, the failure of a single component does not lead to system shutdown (Nguyen et al., 2015). Performance dependencies can be modeled, for example, using a copula framework (e.g. Safaei et al., 2020; Xu et al., 2021). Olde Keizer et al. (2017a) conclude that studies including multiple types of component dependencies treat structural dependence primarily as performance dependence such that the system has series, parallel, series-parallel, or k-out-of-N structure.

Technical dependence means that maintaining a given component either requires or prohibits the maintenance of other components (Olde Keizer et al., 2017a). In practice, this can imply various replacement, disassembly, or usage requirements for maintaining other components. In particular, the effect of disassembly has received attention in the literature. For example, Dao and Zuo (2017) focus on how alternative sequences of disassembly decisions affect the total cost and duration of maintenance. Dinh et al. (2020a) and Dinh et al. (2024) study how disassembly operations affect the degradation processes of components. Dinh et al. (2022) summarize other studies. The review by de Jonge and Scarf (2020) points out that there is a growing need to study technical structural dependencies.

While failed components require corrective maintenance, preventive maintenance actions can be carried out before failure. Identification of the need for such preventive maintenance can be condition-based, predictive, or prognostic (de Jonge and Scarf, 2020). In multi-component systems, it may be useful to maintain several components simultaneously, especially in the presence of positive economic dependence or technical dependence, even if this is not formally required due to component failures. Such opportunistic maintenance can reduce the total number of failures, lower maintenance costs, and increase the lifetime of the system (Ab-Sumat and Kamaruddin, 2014). Still, it can be difficult to determine when the system should be maintained; how the risks of maintaining either too little or too much can be avoided; and how the availability of spare parts and workforce can be ensured in unexpected opportunities for maintenance. Opportunistic maintenance under technical dependency needs more research (Zhou et al., 2015).

Dynamic grouping can be utilized in opportunistic maintenance. In dynamic grouping, groups of components are maintained simultaneously with planned preventive or corrective actions (Chalabi et al., 2016). The groups are typically solved using rolling horizon approaches (Pargar et al., 2017). Wilde-man et al. (1997) present the first rolling horizon approach to minimize maintenance costs by taking advantage of positive economic dependencies between components. This approach has been extended
to include structural dependencies, mostly performance dependencies. Vu et al. (2014) consider systems with positive and negative economic dependencies as well as different configurations of components. Do et al. (2015b) consider availability constraints and limited maintenance teams. Shi et al. (2020) develop a condition-based maintenance framework for a multi-component system consisting of serially connected $k$-out-of-$N$ subsystems and having a system reliability requirement. The work of de Pater and Mitici (2021) introduces a model that builds on monitoring to provide Remaining-Useful-Life prognostics for components. This model – which is solved using rolling horizon integer linear programming – also supports the prognostic-based predictive maintenance planning of multiple multi-component systems with a limited stock of spare parts. In addition to dynamic grouping, rolling horizon approaches can be used, for example, for scheduling trains in a dynamic operation environment Zhang et al. (2021).

Maintenance grouping can also be tackled using techniques such as genetic algorithms (GA) and Particle Swarm Optimization (PSO). Chalabi et al. (2016) apply PSO to improve the system availability, reducing preventive maintenance costs of a series system with positive economic dependence. Wang et al. (2022) propose condition-based maintenance decisions for a multi-component system with economic and stochastic dependencies based on system and component level reliability improvements and optimal groupings that are determined using Monte Carlo simulation and a PSO-based heuristic algorithm. Liang and Parlikad (2020) study predictive group maintenance for multi-system multi-component networks. They introduce a novel GA with an agglomerative mutation that allows the maintenance policy to evolve effectively. Vu et al. (2020) develop a dynamic opportunistic maintenance approach for multi-component redundant systems by using GA with memory. Urbani et al. (2023) formulate a bi-objective optimization problem for a multi-component networked system where flow losses and maintenance costs are minimized using GA. The configuration of the network defines the performance dependencies.

Simulation is applied widely to evaluate the performance of alternative strategies in maintenance scheduling. Zhou et al. (2015) use Monte Carlo simulation to set component-specific preventive maintenance thresholds for a system with structural dependencies. The duration and cost of maintenance depend on the disassembly order of components. Van Horenbeek and Pintelon (2013) assess the performance of heuristic decision rules with simulation. Partial dependencies are modeled with a dependence parameter that accounts for the set-up cost savings associated with the grouping of maintenance activities. Nguyen et al. (2015) use Monte Carlo simulation to assess a system with both economic dependence and performance dependence, while de Pater and Mitici (2021) use simulation to validate the cost savings that their model offers through improved corrective and preventive maintenance for a fleet of aircraft, of which each is equipped with a cooling system consisting of multiple cooling units. Wang et al. (2022) use Monte Carlo simulation to determine optimal groups of components for maintenance.
Maintenance schedules can be established through Markov Decision Processes (MDPs) as well. Love et al. (1982) model vehicles and their conditions as a MDP and determine the best vehicle repair limits with the policy-iteration (PI) algorithm (Howard, 1960). Chan and Asgarpoor (2006) use PI to establish a cost-optimal maintenance policy for a single component that can fail randomly at any time. Kyriakidis and Dimitrakos (2006) model the optimal preventive maintenance of an installation that supplies raw material to a production unit with MDP solved with PI. Wijnmalen and Hontelez (1997) present a heuristic approach to decompose the maintenance scheduling problem into smaller problems, which are solved as MDPs. They include a system-level set-up cost and a type-specific component set-up cost to justify maintenance grouping. Kim and Makis (2009) solve an optimal maintenance schedule of a multi-state, single-component system with two types of failures using a semi-Markov Decision Process and PI algorithm.

MDPs have also been applied to the maintenance scheduling of multi-component systems. Olde Keizer et al. (2016) accommodate economic dependencies in a system in which $k$-out-of-$N$ components must operate for the system to operate. Their solution – which is obtained with dynamic programming – illustrates that the optimal policy outperforms several heuristic policies. In a similar setting, Andersen et al. (2022) assess the computational requirements of different MDP solution methods: value iteration (VI), PI, and modified policy-iteration (MPI). Olde Keizer et al. (2017b) build a MDP for condition-based maintenance and the optimization of spare parts planning for a multi-component system. Olde Keizer et al. (2018) consider a system of identical components with economic dependence and load sharing. Their MPD model shows that in the presence of a strong economic dependence, the optimal policy differs significantly from threshold-based policies. Xu et al. (2021) use a MDP to examine the optimal condition-based maintenance policy under periodic inspection for a $k$-out-of-$N$ system, where economic dependence, stochastic dependence, and imperfect maintenance are emphasized. They apply the VI algorithm with Monte Carlo simulation to determine the optimal inspection interval and the optimal policy. Zheng et al. (2023) use a MDP for joint optimization of condition-based maintenance and spare provisioning. Machado et al. (2023) build a MDP for the preventive maintenance of multi-unit parallel systems by considering a reward function that does not depend on maintenance costs.

Although earlier models capture some system dependencies, only a few of them consider the reliability of the system explicitly or account for technical structural dependencies (see Table 1). Moreover, the technical dependencies addressed are usually limited to disassembly operations while, computationally, the models are solved either with simulation or heuristics with a rolling horizon. We are not aware of optimization approaches for solving cost-efficient maintenance policies when the reliability requirements and the technical structural dependencies must both be accounted for. To fill this gap, we formulate an
3. Model development

In this section, we develop an optimization model to determine a maintenance policy that mini-
mizes the discounted maintenance costs to optimality for a multi-component system over an infinite time
horizon. Possible maintenance actions consist of decisions to replace and disassemble components. Disas-
sembly has no impact on the failure processes and conditions of components, and replacing a component
restores its condition to as-good-as-new. Disassembly may be a mandatory precondition for replacements.
Maintenance actions may involve economic dependencies, meaning that replacing or disassembling one
component can affect the costs of replacing or disassembling other components. The component replace-
ments ensure the reliable operation of the system. Specifically, the system must remain operational until
the next maintenance instance with a probability that exceeds a predetermined reliability threshold.

3.1. System structure, dependencies, and maintenance costs

The system consists of \( n \) components, denoted by the set \( N = \{1, \ldots, n\} \). The operating status of a
component is binary, so a component either operates or does not. The component failure probabilities
follow known probability distributions. There are no stochastic dependencies between components, mean-
ing the components fail independently. The operating status of the system is binary; it either operates
or does not. Every component must operate for the system to operate. Thus, a single component failure
is immediately observed as the system stops operating. We assume that no additional components will
fail when the system has failed.

In practice, multiple operating statuses of a component are often evaluated using a binary distinction
so that a component either performs well enough or is considered failed. This is analogous to common
modeling approaches in which a component is considered failed only when its state reaches some failure
level threshold (see, e.g., Olde Keizer et al., 2018; Xu et al., 2021; Andersen et al., 2022). The challenges
and solution suggestions for introducing additional operating statuses in our setting are listed in Section
5.2.

Components can be disassembled or replaced with as-good-as-new ones during periodic \textit{maintenance
instances} at times \( t_k = k\Delta t, \ k \in \mathbb{N} \), where the constant \( \Delta t > 0 \) is the \textit{maintenance interval}. The
maintenance interval can be measured, for example, using calendar time, working hours, or the distance
that the system has covered. If a component fails during the interval \( (t_k, t_{k+1}) \), it is replaced by a
new one at the next maintenance instance \( t_{k+1} \). This is a reasonable assumption when maintenance
services are a scarce, prebooked resource, for example, when maintenance requires specialized facilities or
advance booking (e.g. Han et al., 2021); when spare parts and expertise are unavailable before the next maintenance instance (e.g. Do et al., 2015a); or when the component causing the system failure is only identified with resources available at maintenance instances (e.g. Maaroufi et al., 2015). Alternatively, the exact timing of future maintenance instances can be guided using rolling horizon approaches (e.g. Wildeman et al., 1997; Baldi et al., 2016), which are effective when maintenance actions are selected only in view of a fixed time horizon. The duration of the component replacements during maintenance instances is assumed to be negligible in comparison with the length of the maintenance interval. These assumptions imply that at most one component can fail during a maintenance interval.

We assume that for every component, we know the maintenance instance the component was replaced the last time. Thus, we know the age of each component at maintenance instance \( t_k \), denoted by \( a_k^i \in \mathbb{R} \), which is a multiple of maintenance intervals. Correspondingly the component ages at maintenance instance \( t_k \) are denoted by the vector \( a^k = (a_k^i)_{i \in N} \). The age of a component is a well-founded proxy measure for its condition. We assume that older components are more likely to fail than newer ones. A binary variable \( f_k^i \) indicates the operating status of component \( i \) at maintenance instance \( t_k \) such that where \( f_k^i = 1 \) if and only if component \( i \) has failed during the maintenance interval \( (t_{k-1}, t_k) \). Correspondingly, the operating status of the system is a binary row vector \( f^k \in \{0, 1\}^{1 \times n} \), which is also referred to as the failure status of the system and which has at most one non-zero element since at most one component can fail during any maintenance interval \( (t_{k-1}, t_k) \). The state of the system is determined by the combination of the ages \( a^k \) and the operating statuses \( f^k \) of all its components. Defining the state space based on component ages offers advantages as information about the components’ ages is usually available and accurate.

Economic and structural dependencies between components are modeled with a directed graph \( G = (V, A) \), where \( V = \{0\} \cup D \cup N \) is the set of nodes and \( A \) is the set of directed arcs \((x, y)\) with start node \( x \) and end node \( y \). The node 0 is the root node, which represents the decision to take maintenance actions. The nodes in \( D \) represent possible component disassembly actions. The nodes in \( N \) represent possible component replacement actions. We assume that the replacement action of each of the \( n \) components of the system is represented by a corresponding node in the graph, i.e., \(|N| = n\).

The root node is associated with a fixed set-up cost \( c_0 \), which is paid whenever any maintenance actions are taken at the maintenance instance (Wildeman et al., 1997). For arc \((x, y)\), there is a weight \( c_{xy} \), which represents the cost of maintenance action \( y \) with the condition that maintenance action \( x \) is also carried out during the same maintenance instance. For example, if a node \( y \in D \cup N \) is connected directly to the root node, the corresponding maintenance action can be carried out independently of other actions, i.e., this action \( y \) does not depend technically on other actions. Otherwise, there is a
path from the root node to node \( y \) that traverses through some other node \( x \), indicating that there is a technical dependence, meaning that this component \( x \) must be either disassembled (if \( x \in D \)) or replaced (if \( x \in N \)) before the maintenance action \( y \) can be carried out. Multiple paths can lead to a single node from the root node. Thus, the maintenance cost of a component can depend on what other components are maintained simultaneously.

In general, corrective replacement is more expensive than preventive replacement (e.g. Dinh et al., 2022). If component \( i \) is replaced due to failure, a component-specific corrective replacement surplus cost \( r_i \geq 0 \) is paid. This cost can represent, for example, the extra cost of using a backup system to provide adequate functional capabilities during the period from the failure event to the next maintenance instance, the extra work required to replace the failed component, or the need to test the system after a component failure.

Concretely, the 5-component system with illustrative cost parameters in Figure 1 helps to illustrate the model. The nodes \( D = \{D1, D2, D3, D4, D5\} \) represent component disassembly options and the nodes \( N = \{1, 2, 3, 4, 5\} \) represent component replacements. There is no arc between the root node and node D2; thus, component 2 cannot be disassembled nor replaced without disassembling and replacing component 1. In practice, this can mean that component 2 is inside component 1, which must be broken down to reach component 2 for its maintenance. Furthermore, it costs less to disassemble component 5 if component 4 is also disassembled during the same maintenance instance because \( c_{D4,D5} = 10 < 50 = c_{0,D5} \). We also assume set-up cost \( c_0 = 60 \) and corrective replacement surpluses \( r = (r_i)_{i \in N} = (120, 90, 85, 70, 90) \) with this maintenance interval. For example, if component 3 has failed and we also replace component 1, the total cost of these maintenance actions is \( c_0 + c_{0,D1} + c_{D1,1} + c_{0,D3} + c_{D3,3} + r_3 = 60 + 30 + 120 + 15 + 75 + 85 = 385 \).

![Figure 1: An example of a system with five components and corresponding costs of \( c_{ij}. \)](image)
3.2. Modeling the evolution of system

The components age and possibly fail during each maintenance interval \((t_k, t_{k+1})\). We assume that the failure time \(t_f^i\) of component \(i\) follows a continuous cumulative distribution function \(\Phi_i\) where \(\Phi_i(a^k_i)\) is the probability of component \(i\) failing before the age \(a^k_i\). If component \(i\) operates at maintenance instance \(t_k\), having age \(a^k_i\), then it operates until \(t_{k+1}\) with the conditional probability

\[
P\left(t_f^i > t_{k+1} | t_f^i > t_k\right) = \frac{1 - \Phi_i(a^k_i + \Delta t)}{1 - \Phi_i(a^k_i)} := \Theta_i(a^k_i),
\]

where \(\Theta_i(x) = 1 - \Phi_i(x)\). This is the reliability of component \(i\) at maintenance instance \(k\) given its age \(a^k_i\) (see Do and Béringuer, 2020). We assume that the reliability of the component decreases as the component ages. Because the system operates only if every component operates and the failures are independent, the reliability of the system at \(t_k\) given component ages \(a^k = (a^k_i)_{i \in N}\) is

\[
R_{\text{sys}}(a^k) = \prod_{i=1}^n R_i(a^k_i),
\]

(1)

We set a reliability threshold \(\rho \in [0, 1)\) for the system. At every maintenance instance, the system is to be maintained so that its reliability in (1) remains above the specified threshold over the entire maintenance interval until the next maintenance instance is reached, i.e.,

\[
R_{\text{sys}}(a^k) \geq \rho
\]

(2)

for all \(k \in \mathbb{N}\).

The probability of no failures occurring is given by (1). The system fails when the first component fails. Component \(i\) will fail first during \((t_k, t_{k+1})\) if its failure time \(t_f^i\) satisfies

\[
t_k \leq t_f^i = \min \{t_1^f, ..., t_n^f\} \leq t_{k+1}.
\]

We denote this event \(E^k_i\). For \(E^k_i\) to occur, all components must have operated until \(t_k\) so that their ages have reached \(a^k\). Now, when the component ages and the maintenance interval are known, the probability of \(E^k_i\) is given by the conditional probability

\[
P(E^k_i(a^k)) = \frac{\int_0^{\Delta t} \left(\prod_{j \neq i} \Theta_j(a^k_j + x)\right) \Phi'_i(a^k_i + x) dx}{\prod_{i=1}^n [\Theta_i(a^k_i)]}
\]

(3)

where \(\Phi'_i(t) = \frac{d}{dt}\Phi_i(t)\) is the probability mass function of the corresponding cumulative distribution function. If the system has only one component \((n = 1)\), then \(P(E^k_i(a^k)) = 1 - R_i(a^k_i)\).
3.3. Maintenance action portfolios

At every maintenance instance $t_k$, a portfolio $z^k \in \{0, 1\}^{1 \times (|D|+n)}$ of maintenance actions is selected. The binary vector $z^k$ indicates whether a maintenance action of node $x$ is carried out ($z^k_x = 1$) or not. The portfolio consists of disassembly decisions $z^{kd} \in \mathbb{R}^{1 \times |D|}$ and replacement decisions $z^{kr} \in \mathbb{R}^{1 \times n}$, and a maintenance action portfolio is $z^k := [z^{kd} \ z^{kr}]$. For example, if we replace components 1 and 4 from the 5-component-system at $t_k$, then $z^{kr} = (1, 0, 0, 1, 0)$. The set of possible maintenance action portfolios is denoted with $Z$.

The component ages are updated between maintenance instances. We get

$$a^k_i = \begin{cases} 0, & \text{if } z^{kr}_i = 1 \\ a^{k-1}_i + \Delta t, & \text{if } z^{kr}_i = 0 \end{cases}$$

for all $i \in N$. We assume that if the system fails due to the failure of some component, then the other components continue to age until the next maintenance instance. This assumption is aligned with the principle of conservatism in risk analysis as it can lead only to biases of over-maintenance because some components may have been in use for less time than what their recorded age in the model would indicate. Moreover, it is noteworthy that components may age even if they are not in use (e.g., due to rusting).

To some extent, the impacts of this assumption are reduced by the fact that the components have increasing failure rates: in other words, the components are more likely to fail towards the end (rather than the start) of the maintenance interval. Computationally, one can mitigate these impacts through shorter maintenance intervals and tighter reliability thresholds. Still, because the MPD algorithm assumes that the maintenance instances occur at prespecified time points, it is not suited for situations in which the age of operational components would not be equal to some multiple of the duration of maintenance intervals.

In general, there are $2^{|D|+n}$ possibilities to choose the maintenance action portfolio at each maintenance instance. However, since maintenance incurs costs, one should not consider portfolios that consist of disassembly decisions only, as these alone do not improve the condition of the system. In addition, the feasible portfolios (Liesiö et al., 2007) at a particular maintenance instance need to satisfy the constraints arising from component failures, the system reliability threshold, and the system structure.

Because a failed component is always replaced, we have

$$z^{kr}_i \geq f^k_i \ \forall i = 1, \ldots, n.$$  

Next, the feasible portfolio must take structural dependencies into account. We define a set

$$\tilde{V} = \{ y \in D \cup N \mid (0, y) \not\in A \}$$
to describe the maintenance action nodes not adjacent to the root node. If these maintenance actions are carried out, there must be a path from the root node to these actions:

$$\sum_{\{i(x,y) \in A\}} z^k_x \geq z^k_y, \quad \forall y \in \hat{V}. \quad (6)$$

This condition highlights the technical structural dependencies where some component replacements are not possible without disassembling or replacing other components. For example, in Figure 1, component 2 cannot be disassembled without disassembling and replacing component 1.

Based on these constraints, we define a feasible maintenance action portfolio.

**Definition 3.1.** A portfolio $z^k$ is called a **feasible maintenance action portfolio** if

1. the updated component ages (4) fulfill the reliability threshold (2),
2. it replaces failed components (5),
3. it satisfies structural dependencies (6).

This framework can even handle other types of feasibility constraints. For example, if at most or at least $b$ components from a set $B \subset N$ must be replaced at the same time, constraints $\sum_{i \in B} z^k_{i^r} \leq b$ or $\sum_{i \in B} z^k_{i^r} \geq b$, respectively, must hold.

For any feasible $z^k$, the corresponding replacement costs can be determined from the graph $G$. This means determining a tree, which connects all maintenance actions in $z^k$ to the root node at the lowest possible cost. Such a tree is a connected subgraph of the graph called a minimum-cost arborescence. This arborescence can be determined using Edmond’s algorithm (Kleinberg and Tardos, 2006) to obtain the corresponding portfolio cost $c(z^k)$. If $z^k \neq 0$, the set-up cost $c_0$ is added in the portfolio cost $c(z^k)$.

Figure 2 shows the minimum cost arborescences of two maintenance action portfolios. With a set-up cost $c_0 = 60$, the cost of replacing components 1 and 5 is $c((1,0,0,0,1,0,0,0,1)) = 60 + 30 + 120 + 50 + 140 = 400$. Adding component 4 to the portfolio increases the cost by 160 to $c((1,0,0,1,1,0,0,1,1)) = 60 + 30 + 120 + 45 + 155 + 10 + 140 = 560$. This increase is smaller than $c_{0,D4} + c_{D4,4} = 200$ because the disassembly of component 5 becomes cheaper when component 4 is disassembled simultaneously.

### 3.4. Formulation as a Markov Decision Process

Under the above assumptions, the system’s maintenance scheduling can be modeled as a **discrete-time Markov Decision Process** (MDP; Howard, 1960). The states of the system form the state space of the MDP. The state is defined by the component ages $a^k$ and the failure status $f^k$, which are known. The state at maintenance instance $t_k$ can be written as $[a^k \ f^k] \in \mathbb{R}^{1 \times 2n}$. Consequently, the state space is

$$S = \left\{ [a^k \ f^k] \mid R_{sys}(a^k) \geq \rho, \sum_{i=1}^{n} f^k_i \leq 1 \right\}.$$
The state space $S$ is finite due to the following two assumptions. First, the reliability threshold requirement (2) for the system reliability and the equation (1) imply that the reliability of each component must fulfill the threshold $\rho$. Because the reliability of a component decreases as the component ages, it will fail to fulfill the reliability threshold unless it is replaced before reaching a certain age. Thus, the number of unique component age combinations that fulfill the reliability threshold (2) is finite. We denote this number with $h$. Second, we assume that at most one component can fail between two consecutive maintenance instances, as explained in Section 3.1. Consequently, only $n + 1$ possible failure statuses of the system exist for every component age combination. As a result, the size of the state space is $|S| = h \cdot (n + 1)$.

At any maintenance instance, the optimal maintenance decisions depend only on the ages of components and their failure status. Thus, we adopt a state representation $\sigma \in S$, where every state $\sigma := [\sigma_a \sigma_f] \in \mathbb{R}^{1 \times 2n}$ is unique, component ages are denoted by $\sigma_a \in \mathbb{R}^{1 \times n}$ and failure status is denoted by $\sigma_f \in \{0, 1\}^{1 \times n}$. In addition, $Z_\sigma \subset Z$ denotes the set of feasible portfolios for state $\sigma$. The action space of the MDP consists of maintenance action portfolios. For the remainder of this section, we consider only feasible portfolios, that is, $z \in Z_\sigma$ for all $\sigma \in S$. The superscript $k$ can be omitted.

Selecting and performing a maintenance action portfolio $z \in Z_\sigma$ updates the current state of the system from $\sigma$ to a post-decision state (Powell, 2007) denoted by $\sigma^z := [\sigma^z_a \sigma^z_f]$. The component ages $\sigma^z_a$ in the post-decision state are updated from $\sigma_a$ using (4). In addition, since failed components are replaced, the failure status in the post-decision state is $\sigma^z_f = 0$. The system then transitions to one of the $n + 1$ states for the next maintenance instance. During the transition, either one of the components fails, or the system stays operational. The transition probabilities are computed with (1) and (3) using the updated component ages $\sigma^z_a$ and given failure distributions $\Phi_i$. More precisely, we denote the state
transition probability from state $\sigma$ to state $\tilde{\sigma}$, when a maintenance action portfolio $z$ is performed, by $p_{\sigma\tilde{\sigma}}(z)$. It is calculated as

$$p_{\sigma\tilde{\sigma}}(z) = \begin{cases} \frac{1}{R_{sys}(\sigma^z)}, & \text{if } \tilde{\sigma}_a = \sigma^z_a + 1 \cdot \Delta t \text{ and } (\tilde{\sigma}_f) = 1 \\ \frac{1}{R_{sys}(\sigma^z)}, & \text{if } \tilde{\sigma}_a = \sigma^z_a + 1 \cdot \Delta t \text{ and } \tilde{\sigma}_f = 0 \\ 0 & \text{otherwise}. \end{cases}$$

\hspace{1cm} (7)

Here $1 \in \mathbb{R}^{1 \times n}$ is a vector of ones. Transition probabilities are positive only if $\tilde{\sigma}_a = \sigma^z_a + 1 \cdot \Delta t$, meaning that after replacements, every component ages with the length of a single maintenance interval. Note that the transition probabilities are stationary, i.e., independent of the maintenance instance $t_k$.

The cost of action $z$ in state $\sigma$, $c_\sigma(z)$, is defined by the cost of the maintenance action portfolio $c(z)$ plus the possible corrective replacement surplus $r = (r_i)_{i \in N}$:

$$c_\sigma(z) = c(z) + \sigma_f r^T.$$ \hspace{1cm} (8)

3.5. Solving the model with modified policy-iteration

A policy $U: S \rightarrow Z$ is a function that prescribes a feasible maintenance action portfolio $z$ when the system is in state $\sigma$: $U(\sigma) = z$. We focus on stationary policies that do not depend on the current maintenance instance $t_k$. The optimal maintenance action portfolio for each possible state is given by the optimal policy that minimizes the net present value of the maintenance costs over a long time period.

The long-term costs of a discounted MDP with policy $U$ are represented by a value vector $v^U \in \mathbb{R}^{|S|}$ where $v^U(\sigma)$ stands for the long-term expected cost of the maintenance when the system is currently in state $\sigma$. The value vector is a solution to the Bellman equation

$$v^U(\sigma) = c_\sigma(U(\sigma)) + \beta \sum_{\tilde{\sigma} \in S} p_{\sigma\tilde{\sigma}}(U(\sigma)) v^U(\tilde{\sigma})$$

where $0 \leq \beta < 1$ is the discount factor.

The value vector and the optimal policy are solved with a Anderson accelerated Gauss-Seidel Modified Policy Iteration Algorithm (AAGSMPI). The algorithm uses the Gauss-Seidel Modified Policy Iteration Algorithm (Puterman, 2014) where the value computation is Andersson accelerated (Anderson, 1965). The Gauss-Seidel value computation is defined as an operator $v_{n+1} = T_{GS}^n v_n$, where

$$v_{n+1}(\sigma) = c_\sigma(U(\sigma)) + \beta \sum_{\tilde{\sigma} < \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_{n+1}(\tilde{\sigma}) + \beta \sum_{\tilde{\sigma} \geq \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_n(\tilde{\sigma})$$ \hspace{1cm} (9)

for all $\sigma \in S$. Similarly, the Gauss-Seidel ValueIteration is defined as an operator $v_{n+1} = T_{GS} v_n$, where

$$v_{n+1}(\sigma) = \min_{U(\sigma) \in Z_\sigma} \left\{ c_\sigma(U(\sigma)) + \beta \sum_{\tilde{\sigma} < \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_{n+1}(\tilde{\sigma}) + \beta \sum_{\tilde{\sigma} \geq \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_n(\tilde{\sigma}) \right\}$$ \hspace{1cm} (10)
for all $\sigma \in S$. Here, the state space $S$ is looked at as an indexed list where $\hat{\sigma} < \sigma$ is true for every state $\hat{\sigma}$ that has a smaller index than $\sigma$. Therefore, the updates are done in order, from the state with the smallest index to the largest.

Next, Anderson acceleration is introduced. Let $U$ be the chosen policy and $T^U_{GS}$ the value computation operator introduced previously. First, the current estimate of value vector $v_m$ is calculated, and the last $k$ previous estimates $v_{m-1}, \ldots, v_{m-k}$ for the value vector $v^U$ are memorized. Here, $k$ represents the size of the memory. Then, a vector $\alpha \in \mathbb{R}^{k+1}$ is defined by

$$
\alpha = \arg \min_{\alpha \in \mathbb{R}^{k+1}} \left\| \sum_{i=0}^{k} \alpha_i (T^U_{GS}v_{m-k+i} - v_{m-k+i}) \right\| \text{ st. } \sum_{i=0}^{k} \alpha_i = 1.
$$

Now, the new estimate $v_{m+1}$ can be calculated by

$$
v_{m+1} = \sum_{i=0}^{k} \alpha_i T^U_{GS}v_{m-k+i}.
$$

In equation (11), different norms can be used. We apply the $L_2$ norm since the optimization problem (11) can be solved analytically using the Karush-Kuhn-Tucker conditions. We define $B_i = T^\pi_{GS}v_i - v_i \in \mathbb{R}^{|S|}$ and $B = [B_{m-k} B_{m-k+1} \ldots B_{m-1} B_m] \in \mathbb{R}^{|S| \times (k+1)}$. According to Geist and Scherrer (2018), we can now calculate the vector $\alpha$ by

$$
\alpha = \left( B^T B \right)^{-1} \mathbf{1} \left( B^T B \right)^{-1} \mathbf{1},
$$

where $\mathbf{1} \in \mathbb{R}^{k+1}$ refers to a vector of ones. Problem (12) can be prone to numerical instability, which can be alleviated using regularization techniques (Park et al., 2022).

The steps of the AAGSMPI Algorithm consist of the initialization step and the repeated steps of policy improvement and partial value evaluation with an $\varepsilon$-optimal stopping criterion. Details are presented in Algorithm 1. In the algorithm, $M$ is the number of value computation iterations, and Anderson acceleration is conducted only once in each iteration round using $k = M$ as the size of the memory.
Initialization Formulate the directed graph. Determine the portfolio costs $c(z)$ using Edmond’s algorithm. Formulate state space $S$ and store it in memory. Calculate the needed state transition probabilities $p_{\sigma\tilde{\sigma}}(z)$ and store them in memory. Select $v_0(\sigma)$ for all $\sigma \in S$, specify $\varepsilon > 0$, choose $M \in \mathbb{N}$.

for $n \geq 0$ do

Policy improvement

for $\sigma \in S$ do

Set $u^0_n(\sigma) = \min_{U(\sigma) \in \mathcal{Z}_\sigma} \left\{ c_\sigma(U(\sigma)) + \beta \sum_{\tilde{\sigma} < \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_{n+1}(\tilde{\sigma}) + \beta \sum_{\tilde{\sigma} \geq \sigma} p_{\sigma\tilde{\sigma}}(U(\sigma)) v_n(\tilde{\sigma}) \right\}$.

Set $U_{n+1}(\sigma)$ to the argument $U(\sigma)$, which minimizes the right-hand side of the above equation, which is analogous to (10).

end

if $|u^0_n(\sigma) - v_n(\sigma)| < \varepsilon(1 - \beta)/2\beta$ \(\forall \sigma \in S\) then

return $U_{n+1}$

end

Anderson accelerated partial value computation

for $m$ from 0 to $M$ do

$u^m_{m+1} = T^{U_{m+1}}_{GS} u^m_m$ (every state updated in order using equation (9))

$B_m = u^m_{m+1} - u^m_m$

end

Set $B = [B_0 \ B_1 \ ... \ B_{M-1} \ B_M]$

Calculate $\alpha$ with (12)

Set $v_{n+1} = \sum_{i=0}^{M} \alpha_i T^{U_{m+1}}_{GS} u^i_m$

end

Algorithm 1: Anderson accelerated Gauss-Seidel Modified Policy Iteration Algorithm consists of the initialization step, and the repeated steps of policy improvement and Anderson accelerated partial value computation.

4. Illustrative example

4.1. Problem description

We illustrate the maintenance scheduling model with an example of ground transportation equipment. This system consists of four components, engine 1 (E1), engine 2 (E2), chassis (C), and wheels (W), which deteriorate over time and have technical structural dependencies. The components must be disassembled
before they can be replaced. Also, before disassembling the chassis, both engines must be disassembled. The chassis (and the engines) must be disassembled to disassemble the wheels. Economic dependence of the system is positive: for every maintenance operation there is a fixed set-up cost $c_0 = 388$ with maintenance interval $\Delta t = 1$, and some disassembling costs can be avoided if multiple components are maintained simultaneously.

Component failures are assumed to follow the Weibull distribution. The component-specific shape parameters $k$, scale parameters $\lambda > 0$, and maintenance costs are in Table 2. They reflect data from a real-world problem. Engines 1 and 2 follow identical failure distributions, but Engine 2 is slightly more expensive to replace. For all distributions, the shape parameter $k > 1$ means that older components are more likely to fail than younger ones.

Table 2: Maintenance costs and failure distributions of different components.

<table>
<thead>
<tr>
<th>component</th>
<th>abbr.</th>
<th>component specific maintenance costs</th>
<th>Weibull parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>disassembly</td>
<td>replacement</td>
</tr>
<tr>
<td>Engine 1</td>
<td>E1</td>
<td>23</td>
<td>393</td>
</tr>
<tr>
<td>Engine 2</td>
<td>E2</td>
<td>28</td>
<td>403</td>
</tr>
<tr>
<td>Chassis</td>
<td>C</td>
<td>167</td>
<td>413</td>
</tr>
<tr>
<td>Wheels</td>
<td>W</td>
<td>0</td>
<td>1000</td>
</tr>
</tbody>
</table>

Figure 3 presents the system as a directed graph based on the data in Table 2. This graph includes both disassembly and replacement decision with $D = \{DE1, DE2, DE12, DC, DW\}$ and $N = \{E1, E2, C, W\}$. The node ‘DE12’ is the decision to disassemble both engines, which is a prerequisite for replacing either the chassis or the wheels.

The graph in Figure 3 can be presented in a simplified form. Some disassembly nodes have been merged with the corresponding replacement nodes because it would make sense to disassemble a component only if it results in replacement(s). This gives the simplified graph in Figure 4 with $D = \{DE12\}$ and $N = \{E1, E2, C, W\}$. For example, when both engines are disassembled, the replacement cost of wheels is equal to the sum of the wheel replacement cost (1000) and the cost of disassembling the chassis (167). Thus, in Figure 4 we have $c_{DE12,W} = 1000 + 167 = 1167$. 

---

19
Next, the optimal maintenance scheduling policy is solved with AAGSMPI for a maintenance interval of $\Delta t = 1$, reliability threshold $\rho = 0.80$, tolerance $\varepsilon = 1$, number of value computations $M = 8$, and discount factor $\beta = 0.99$. With the given Weibull distributions, the number of different component age combinations that satisfy the reliability threshold is $h = 3104$. This value is obtained by systematically testing different $a^k$s in equation (2). Consequently, the size of the state space is $|S| = 3104 \cdot (4 + 1) = 15520$. The maintenance scheduling problem is solved with MATLAB. The state transition probabilities solved with equation (3) are calculated with numerical integration.

4.2. Optimal Policy

The optimal maintenance scheduling policy is solved using AAGSMPI. To initialize the algorithm, $v_0(\sigma)$ is selected as the cheapest feasible maintenance action portfolio cost for each $\sigma \in S$. The optimal policy is obtained with $v_{17}$ in approximately 5.2 seconds with a standard computer (Intel i5-1135G7 CPU, 2.40GHz, 16GB of RAM). For any of the 15520 states, this policy tells which components should
be disassembled and replaced at the maintenance instance, given the failure statuses and ages of all the components. The results also help estimate the total discounted maintenance cost for the system during an infinite time horizon. After the final iteration, this estimate is obtained directly from the $v^U$ values. For example, for a brand-new system $\sigma_n$ with a discount factor $\beta = 0.99$, this cost is $v^U(\sigma_n) \approx 49820$.

For comparison, the optimal policy is also calculated using the policy iteration algorithm in Appendix A. The system of linear equations (A.1) is solved approximately using the conjugate gradients squared method (Sonneveld, 1989). The optimal policy is obtained in approximately 48 seconds, which is more than nine times longer than with AAGSMPI. Both algorithms give the same optimal policy.

The optimal maintenance scheduling policy provides component-specific disassembly and replacement suggestions for each possible state. Explaining these recommendations in a compact form for future states can be difficult because of the large number of these states (for example see Xu et al., 2021; Zheng et al., 2023). Nevertheless, the number of states can be reduced because, for a brand-new system, not all states are reachable if the system is maintained according to the optimal stationary maintenance scheduling policy. When the system is maintained with the optimal maintenance action portfolios, the MDP turns into a Markov chain where the system state transitions between the reachable states. Specifically, by repeatedly applying this Markov chain’s transition probability matrix to any initial state, we can determine the steady state distribution and identify the reachable states. For example, only 4005 of the 15520 states are reachable with the parameters in this illustrative example.

Figure 5: Ages, at which working components are replaced under optimal maintenance policy when there are no component failures (left) or wheels have failed (right). Green denotes that the component is never replaced at the indicated age. Yellow denotes that the component is replaced under some combination of other components’ ages. Red denotes that the component is always replaced at the indicated age.
Leppinen et al.

Figure 5 illustrates component-specific preventive replacement recommendations as a function of the component ages, considering reachable states in which either no components have failed (left), or the wheels have failed (right). The cells summarize information based on recommendations for the reachable states, conditioned on the age of one component. For example, in the left pane with no component failures, the recommendation for all reachable states is that wheels of age 2 should not be replaced; this cell is green. On the other hand, for wheels of age 4, there are some states for which the optimal maintenance action portfolio recommends replacing the wheels; hence, this cell is yellow.

Replacing the wheels requires disassembling the engines and the chassis. Furthermore, wheels are the most expensive, and engine 1 is the least expensive component to be replaced. These characteristics are reflected in the decision recommendations. Without failures, wheels replacements start at age four, which is later than for other components. On the other hand, replacing engine 1 helps keep the system reliability above the reliability threshold: engine 1 can be replaced already at the age of two as a cheap option. Such information is useful for resource planning.

When wheels fail, the mandatory disassembly of engines and chassis, preceding the corrective replacement of wheels, creates a cost-efficient opportunity to preventively replace other components as well: for example, it is always optimal to replace both engines and chassis already at the age of four. For some states, replacing engine 2 at the age of two is optimal, unlike in the case of no failures. Other component failures can be illustrated similarly to explain how different component failures influence optimal maintenance decisions.

Because each state has an optimal maintenance action portfolio, the steady state distribution indicates how likely these different portfolios are optimal decisions over time. In Table 3, these probabilities are conditioned on the five possible failure statuses. To illustrate the effect of the reliability threshold $\rho$, the results for optimal replacement decisions are given for $\rho = 0.80$ and $\rho = 0.83$. With $\rho = 0.83$, 4480 of the 12630 states are reachable for a brand-new system. For example, with $\rho = 0.80$, chassis failure occurs with probability $\approx 0.9\%$ (the corresponding column sum), and in this case, the optimal policy almost always suggests replacing every component of the system.
Table 3: Probabilities of alternative maintenance action portfolios depend on the failure status of the system and the reliability threshold (in this case $\rho = 0.80$ or $\rho = 0.83$). The portfolio $z^k = [z_{kd} \ z_{kr}] = [z_{dE12} \ z_{rE1} \ z_{rE2} \ z_{rC} \ z_{rW}]$ includes the disassembly and replacement decisions.

<table>
<thead>
<tr>
<th>portfolio $z^k$</th>
<th>Engine 1 fails</th>
<th>Engine 2 fails</th>
<th>Chassis fails</th>
<th>Wheels fail</th>
<th>no failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,1,0,0,0)</td>
<td>0.1%</td>
<td>1%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(0,0,1,0,0)</td>
<td>-</td>
<td>-</td>
<td>0.1%</td>
<td>1%</td>
<td>-</td>
</tr>
<tr>
<td>(0,1,1,0,0)</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,0,0,1,0)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,0,0,0,1)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,1,0,0,1)</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,0,1,0,1)</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,0,1,1,1)</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,1,1,0,1)</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>-</td>
</tr>
<tr>
<td>(1,1,0,1,1)</td>
<td>&lt;0.1%</td>
<td>0.4%</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>0.1%</td>
</tr>
<tr>
<td>(1,0,1,1,1)</td>
<td>-</td>
<td>&lt;0.1%</td>
<td>0.4%</td>
<td>&lt;0.1%</td>
<td>1.1%</td>
</tr>
<tr>
<td>(1,1,1,1,1)</td>
<td>0.6%</td>
<td>&lt;0.1%</td>
<td>0.6%</td>
<td>&lt;0.1%</td>
<td>0.9%</td>
</tr>
<tr>
<td>(0,0,0,0,0)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>82.5%</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.80</td>
<td>0.83</td>
<td>0.80</td>
<td>0.83</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 3 shows noteworthy results. For $\rho = 0.80$, there is a probability $99.6\% = 0.6\% + 0.6\% + 0.9\% + 2.9\% + 12.1\% + 82.5\%$ that the optimal policy is either $z_{kr} = (0,0,0,0)$ or $z_{kr} = (1,1,1,1)$. Thus, the optimal policy is almost always a block replacement policy such that either no component or every component is replaced. In contrast, the higher reliability threshold $\rho = 0.83$ places stricter requirements on the system, increasing the need for preventive maintenance and decreasing the probability of doing no maintenance. Furthermore, based on the last column of Table 3, it is never optimal to replace all components preventively: the probability of the replacement decision $(1,1,1,1)$ is zero. For most maintenance instances in which maintenance actions are carried out, the optimal policy either replaces engine 1 or 2 only or replaces all components except one engine. This result may seem non-intuitive because the system is maintained more often than with the block replacement heuristic. This illustrates that optimization is needed to group replacements cost-efficiently when opportunities arise.

Finally, adding the probabilities of “no failures” in the columns of Table 3 indicates how often the
system is maintained without failures. For $\rho = 0.80$, this is the case 94.7\% of the time, and for $\rho = 0.83$, this occurs 93.7\% of the time: in other words, the system rarely fails. Equivalently, it is rare that operational components continue to age in a system that has failed. Furthermore, given Table 3, it is unlikely that the optimal policy will replace the failed component only because other components are replaced preventively simultaneously.

4.3. Comparison with heuristic policies

We next compare the optimal policy to two simple heuristic policies: minimal replacement policy and block replacement policy. In the minimal replacement policy, the least costly feasible portfolio is always chosen, which emphasizes replacing only failed components if the reliability threshold is otherwise fulfilled. Block replacement policy replaces all components if one of the components has failed or the system fails to satisfy the reliability threshold otherwise.

We analyze the costs of three alternative policies by examining the impacts of possible component failures through Monte Carlo simulation. In one simulation round, the system is maintained based on the selected policy for 150 consecutive maintenance intervals. The discount factor $\beta = 0.99$ and maintenance interval $\Delta t = 1$ are fixed, but the reliability threshold varies between 0.75 and 0.95. For each reliability threshold, 10,000 simulation rounds are carried out. The average cumulative maintenance costs of the three policies are in Figure 6 for two different set-up costs $c_0 = 388$ and $c_0 = 194$. 

24
Figure 6 presents several notable results. First, the relative superiority between the minimal and the block replacement policies depends on the parameters. For example, the minimal replacement policy is better than the block replacement policy with $c_0 = 194$ and $\rho = 0.91$. Second, the costs of the block replacement policy resemble a step function in shape; indeed, the policy depends only on the reliability threshold. For example, if all components are operational and have been in use for four maintenance intervals, then, based on the assumed failure distributions, the system will continue to operate with probability 0.905 until the next maintenance instance. Thus, preventive maintenance is not required if the reliability threshold is lower than 0.905.

Third, the difference between block replacement and optimal replacement costs depends on the reliability threshold. It is cost-efficient to depart from the block replacement policy when $\rho = 0.83$, for example. In this case, portfolios that replace only one engine are optimal for some states (see the right-most column of Table 3). In addition, with optimal policy, lowering the reliability threshold always leads to lower costs. However, increasing the threshold does not increase the costs dramatically, which is not true for the block replacement policy.
Figure 6 also shows how the total cost of the optimal policy depends on the reliability threshold value. For example, choosing between the reliability thresholds 0.90 and 0.91 impacts maintenance costs more than choosing between 0.89 and 0.90.

On the other hand, for some reliability thresholds, the block replacement policy performs almost as well as the optimal one. For example, in Table 3, the optimal policy with \( \rho = 0.80 \) behaves mostly like the block replacement policy. However, simulations show that these policies differ in states that are reached with low probabilities. Thus, the differences in overall maintenance costs can be minor with some reliability thresholds. The benefits of the optimal policy over the block replacement policy are more pronounced when the set-up cost is lower because replacing a single component or a few components becomes cheaper.

4.4. Computational illustrations

The computational complexity of AAGSMPI depends on the size of the state space. The size depends primarily on the reliability threshold \( \rho \) and the length of the maintenance interval \( \Delta t \) because lowering the threshold or shortening the maintenance interval increases the number of different component age combinations that fulfill the threshold, denoted by \( h \). While the decision maker should set the reliability threshold to reflect the organization’s operational environment and preferences, the choice of maintenance interval can also be a computational issue. Short maintenance intervals can model maintenance operations more realistically and offer flexibility in implementing maintenance plans. Still, they can also give rise to large state spaces. Figure 7 illustrates how the run times of the AAGSMPI algorithm increase with the size of the state space. The circles on the left before the green background indicate run times of the same 21 optimal policies in Figure 6. The five circles on the green background represent additional calculations with larger state spaces corresponding to reliability thresholds lower than 0.75. In each case, the initialization step is the most time-consuming step, which is reasonable since state transition probabilities are solved with numerical integration, which takes time. Overall, the run times increase quite linearly.
Another computational consideration is presented next. To relax the assumption that the system cannot be maintained before the next maintenance instance is reached, we reconsider the example presented in Section 4.1 by considering a shorter maintenance interval $\Delta t = 0.5$. As a result, maintenance actions can be done twice as often. The discount factor is adjusted to $\beta = \sqrt{0.99}$ to compare the long-term costs with the ones obtained in Section 4.2. Reliability threshold $\rho = 0.80$, tolerance $\varepsilon = 1$, and the number of value computations $M = 8$ are kept the same. Since the maintenance interval is shortened, the set-up cost is increased by $30\%$, and the corrective replacement surpluses are decreased by $50\%$. These changes lead to $c_0 = 504.4$ and $r = (150, 150, 80, 306.5)$.

With the given Weibull distributions, the number of different component age combinations that satisfy the reliability threshold is now $h = 101630$. This value is obtained by systematically testing different $a^k$s in equation (2). Consequently, the size of the state space is $|S| = 101630 \cdot (4 + 1) = 508150$, which is over 32 times larger than with $\Delta t = 1$. The maintenance scheduling problem is solved using AAGSMPI.
Leppinen et al.

with MATLAB. The state transition probabilities solved with equation (3) are calculated with numerical integration, and \( v_0(\sigma) \) is again selected as the cost of the cheapest feasible maintenance action portfolio for each \( \sigma \in S \). The initialization step took 135 seconds using a standard computer (Intel i5-1135G7 CPU, 2.40GHz, 16GB of RAM). After the initialization step, the optimal policy is obtained with \( v_{353} \) in approximately 154 seconds. Thus, this much larger problem is still solvable within minutes. The problem of this size is not realistically solved using a policy iteration algorithm (Appendix A) with a standard computer. For a brand-new system, the total discounted maintenance cost during an infinite time horizon is 46117, which is smaller than the cost obtained with \( \Delta t = 1 \), despite the set-up cost increase.

5. Discussion and model extensions

Our model supports decisions on many levels, ranging from operational and tactical considerations to strategic levels (see e.g. Marquez and Gupta, 2006). The optimal policy supports operational planning by specifying what replacements need to be performed at any maintenance instance, depending on each component’s age and failure status, to ensure the reliable and cost-efficient operation of the system. Tactical level planning is demonstrated by the example in Section 4. In particular, results such as those in Figure 5 help anticipate the need for maintenance actions in the near future, making it easier to ensure the reliability of daily services provided by the system (e.g. Baldi et al., 2016; Zhang et al., 2019; Lin et al., 2023; Ji et al., 2024). Once the optimal policy has been determined, it can be used to simulate the expected number of required component replacements, which, in turn, allows managers to better anticipate how many spare parts will be needed over a given time horizon, which helps them order these spare parts in time to avoid delays in completing maintenance actions.

At the level of strategic planning, the model can be used to explore how the expected maintenance costs vary when the reliability threshold, the length of the maintenance interval, or both are changed. Such exploratory analyses can guide the selection of the reliability threshold or the length of the maintenance interval (or both) if the decision maker can choose these parameters. However, the level of the reliability threshold often depends on the requirements of the application context. For example, safety-critical systems necessitate a high reliability threshold. At the limit, if the reliability threshold approaches one and all components deteriorate with aging, the whole system would be replaced by the policy-iteration algorithm at each maintenance interval. At the other extreme, if the reliability threshold approaches zero, the model allows the system to fail even with a high probability. Consequently, very little preventive maintenance will be carried out. For much of the time, the system would, therefore, not be operational for its intended use, which could give rise to significant costs that correspondingly high corrective replacement surpluses could capture. In effect, these trade-offs could be considered in seeking to set a reliability
threshold that represents an appropriate balance between the costs of low performance, on the one hand, and the costs of maintenance, on the other hand. As illustrated by Figure 6, changing the reliability threshold can have either a major or a minor influence on the maintenance costs of the optimal policy.

To explore the cost impacts associated with the length of maintenance interval, one can, for instance, assign a higher set-up cost \( c_0^{\Delta t_1} > c_0^{\Delta t_2} \) if the higher maintenance readiness represented by a shorter maintenance interval \( \Delta t_1 < \Delta t_2 \) gives rise to higher costs. In addition, the corrective replacement surplus \( r_i \) may depend on the length of the maintenance interval. For shorter maintenance intervals, the expected costs of failure can be expected to be lower (i.e., \( \Delta t_1 < \Delta t_2 \) implies \( r_i^{\Delta t_1} \leq r_i^{\Delta t_2} \)) because the next maintenance instance tends to come sooner after failure. We addressed this in a computational illustration of Section 4.4.

Furthermore, by varying technical parameters, such as replacement costs or failure probabilities, we gain insights into how sensitive the optimal policy is to these parameters. This analysis helps evaluate the importance of accurate estimates for these parameters, emphasizing that failure probabilities should not be treated as fixed values. Even when maintenance would not be possible before the next instance, the exact monitoring of failure times offers valuable insights into the reliability of aging components so that their failure probability estimates can be improved.

Finally, the model can also be used to support system design. For example, by adding new edges or removing existing ones from the directed graph, one can examine how structural dependency changes would affect maintenance costs. As a concrete example in the graph in Figure 3, one can add an extra edge from the root node directly to the DW node to explore the impacts of disassembling and replacing wheels without disassembling other components.

5.1. Extensions for system dependencies and condition-based maintenance

Our results open up directions for further development. Failure distributions could be studied by allowing for structural dependencies in the degradation of components (see, e.g., Dinh et al., 2024). In addition, the trade-off between cost and reliability can be approached with methods of multi-criteria decision-making to make selections among alternative maintenance strategies (e.g. Liang et al., 2023).

In addition, the model could be adapted to the problems of condition-based maintenance, which has received a lot of attention recently (e.g. Olde Keizer et al., 2017a; Shi et al., 2020; Xu et al., 2021; Wang et al., 2022; Zheng et al., 2023). In general, condition-based maintenance builds on prognostic that provides failure predictions before failures occur (Jardine et al., 2006). In our current model, such prognostics consist of reliability estimates that have been derived from the components’ lifetime failure models. Thus, to exploit information about the observed condition of components, one may introduce more accurate degradation models (Do and Béreguer, 2020), possibly together with lifetime failure
models (see, e.g., Kivanç et al., 2024). Data on the condition of the components could be obtained by inspecting components during maintenance instances or by monitoring components during maintenance intervals. Then, data-driven approaches can be used to predict possible failures of the components more accurately than with lifetime failure models. This is highly valued in reliability-centered businesses such as railroad transportation (Mohammadi et al., 2021). In addition, such analyses can be helpful from the reverse perspective: that is, by computing optimal scheduling policies with different component failure distributions, it is possible to assess whether inspecting or monitoring these components pays off.

5.2. Extensions to accommodate multiple operating statuses

In the proposed model, we assume a binary representation of operating statuses for both system components and the system itself. The use of multiple operating statuses along with information about the components’ ages could offer advantages (i) if such statuses can be observed accurately so that they can be employed as a basis for developing maintenance policies and (ii) information about these operating statuses help predict the reliability of the system at the next maintenance instance better than what would be possible based on the components’ ages alone.

Assuming that these conditions hold and that maintenance action portfolios impact the reliability of the system through component replacements, our model can be extended to admit multiple operating statuses as follows:

- The state of the system can be represented by extending the vector $\sigma = \left[ \sigma_a, \sigma_f \right]$ so that the $i$-th element of $\sigma_f$ indicates the operating status of component $i$ which now has more than two possible values.

- If the action portfolio $z$ is implemented when the system is state $\sigma$, it is possible that the system may contain components whose status is other than operational or failed. Thus, the reliability of the system no longer depends solely on the ages of its components but also on their operating statuses. In consequence, one has to estimate the reliability $R_{sys}(\sigma^z_a, \sigma^z_f)$ with which the system will function satisfactorily at the next maintenance instance, given that the portfolio $z$ has been applied to the state $\sigma$. In particular, the portfolio $z$ will be feasible if the $R_{sys}(\sigma^z_a, \sigma^z_f) \geq \rho$.

- If portfolio $z$ is feasible when the system is in state $\sigma$, the transition probabilities $p_{\sigma \bar{\sigma}}(z)$ in equation (7) need to be estimated to characterize how likely the system is to be in state $\bar{\sigma}$ at the next state maintenance instance. These probabilities must be aligned with the replacement decisions so that the components’ ages continue to evolve in line with equation (4).
While the terms $R_{\text{sys}}(\sigma^a_z, \sigma^f_z)$ and $p_{\sigma \tilde{\sigma}}(z)$ appear simple, it can be challenging to estimate them. In particular, the reliability of the system $R_{\text{sys}}(\sigma^a_z, \sigma^f_z)$ can no longer be derived from the reliability of its components using equations (1) and (3), which encode the assumption that the system works as long as all its components are operational. Instead, one would have to assess, for example, with what probability the system would continue to function until the next maintenance instance for any combination of ages and operating statuses of its components. Furthermore, the estimation of the state transition probabilities $p_{\sigma \tilde{\sigma}}(z)$ presumes that there is enough data on components of different ages to estimate the probabilities with which they will be in different operating statuses.

If operating statuses cannot be observed, they cannot be employed as one of the inputs that are employed in implementing maintenance policies. For such settings, one could build Partially Observable Markov Decision Processes or develop influence diagrams to represent how information about the components’ underlying operating statuses is mediated to inform maintenance decisions and, moreover, how the quality of this information can be improved. For example, Mancuso et al. (2021) optimize maintenance policies for an industrial problem with a sensor and turbine whose operating statuses cannot be observed with certainty.

5.3. Extensions to more frequent and instantaneous component replacements

In the proposed model, failed components are not repaired immediately but, rather, only when the next maintenance instance is reached. If components are to be replaced instantaneously, then continuous-time models are needed to represent the problem exactly. It is plausible to require that replacements during the maintenance intervals cost more than those at the predefined maintenance instances (otherwise, it would not be optimal to invest in preventive maintenance at all, as there would be no cost penalty for allowing components to be used until they fail). Also, preventive replacements would not be made unless older components are more prone to failure. The very rationale for preventive replacements of operational components at maintenance instances is that they reduce the expected number of instantaneous (and therefore more costly) replacements between maintenance instances.

To illustrate this setting, consider a simple choice between preventive replacement and instantaneous corrective replacement of an operational component of a single-component system over a single maintenance interval of length $\Delta t$. If the component failure time follows the cumulative probability distribution $\Phi(\cdot)$, the expected preventive replacement cost $C_P(a; t)$ for an operational component of age $a$ at the start of the maintenance interval and time $t$ remaining before the end of the maintenance interval, and the expected costs of instantaneous corrective replacements $C_I(a; t)$ can, in principle, be solved from the
non-trivial integral equations

\[ CP(a; \Delta t) = c_P + \int_0^{\Delta t} \left[ c_I + CI(0; \Delta t - \delta)\right] \Phi'(\delta) d\delta \]

\[ CI(a; \Delta t) = \int_0^{\Delta t} \left[ c_I + CI(0; \Delta t - \delta)\right] \Phi'(a + \delta) d\delta \]

where \( c_P \) and \( c_I \) (which is greater than \( c_P \)) are the corresponding costs of preventive and instantaneous replacements, respectively (for more details, see, e.g. Barlow and Proschan, 1996). In this case, the preventive replacement is chosen if \( CP(a; \Delta t) < CI(a; \Delta t) \). The equations for multi-component systems for multiple periods would be even more daunting and are not amenable to the use of efficient algorithms for Markov Decision Processes.

Finally, we have assumed that if the system fails due to the failure of some component, then the other components continue to age until the next maintenance instance. If this assumption does not hold, meaning that the system has components that do not deteriorate after the system has failed, the accuracy of our discrete-time model can be improved if the time of failure is known. Specifically, information about the system’s failure can be used to assess how long the non-deteriorating components have been part of an operational system. Then, one may proceed by evaluating strategies in which the age of these non-deteriorating components is approximated from below (above) by how many periods at least (at most) it has been in operational use. The optimal policy can be solved for these approximations to derive recommended maintenance actions as usual. In particular, the recommendations that are based on the upper bounds are guaranteed to fulfill the reliability threshold.

6. Conclusion

In this paper, we have tackled several recently identified needs in developing maintenance scheduling models (see de Jonge and Scarf, 2020) by developing an optimization model for a multi-component system whose economic and technical structural dependencies are represented by a directed graph. The failure probability of the system is constrained by a reliability threshold to ensure the sufficiently reliable and safe operation of the system. The system is modeled as a Markov Decision Process, and the optimal maintenance decisions are computed with an AAGSMPI algorithm. The model is demonstrated with a realistic case example consisting of four components and their disassembly and replacement decisions. The optimal policy is calculated, analyzed, and compared to block and minimal replacement policies using Monte Carlo simulation. The optimal policy is shown to outperform these two heuristics for all model parameters. It also suggests cost-efficient options for grouping component replacements while ensuring compatibility with the reliability threshold. Finally, some computational illustrations and run times of the algorithm are presented.
Acknowledgements

The work of Jussi Leppinen has been partly funded by Shift2Rail Joint Undertaking (JU) under grant agreement No 101015423.

The work of Tommi Ekholm has been carried out with funding from the Academy of Finland (decision number 311010).

Appendix A. Policy-iteration algorithm

A stationary policy can also be solved using the policy iteration algorithm Howard (1960). The steps of the algorithm are:

1. **Initialization:** Formulate the directed graph of the system. By using Edmond’s algorithm, determine the total costs $c(z)$ of different maintenance action portfolios. Choose some stationary policy $U$.

2. **Value-determination step:** For the current policy $U$, compute the unique solution $v^U \in \mathbb{R}^{|S|}$ to the following system of linear equations:

$$
    v^U(\sigma) = c_\sigma(U(\sigma)) + \beta \sum_{\bar{\sigma} \in S} p_{\sigma\bar{\sigma}}(U(\sigma))v^U(\bar{\sigma}), \quad \sigma \in S. \tag{A.1}
$$

3. **Policy-improvement step:** For each state $\sigma \in S$, determine an updated stationary policy $U'$ by setting

$$
    U'(\sigma) = \arg \min_{z \in Z_\sigma} \left\{ c_\sigma(z) + \beta \sum_{\bar{\sigma} \in S} p_{\sigma\bar{\sigma}}(z)v^U(\bar{\sigma}) \right\}. \tag{A.2}
$$

4. **Convergence test:** If the new policy $U'$ equals $U$ the algorithm is stopped with policy $U$ being optimal. Otherwise, set $U = U'$ and go to step 2.

The algorithm proceeds accordingly as follows. First, a stationary policy is initialized, for example, by choosing the cheapest maintenance portfolio (evaluated with equation (8)) from feasible portfolios $Z_\sigma$ for every state $\sigma \in S$. Second, the values of (A.1) are determined by calculating $v(U)$ from the linear system in matrix form $v^U = c + \beta P v^U$. This system has a unique solution $v^U = (I - \beta P)^{-1}c$ (Puterman, 2014). As a remark, $v^U(\sigma)$ is the expected total discounted maintenance cost of the policy $U$ when the system is currently in state $\sigma$.

Third, the expression on the right side of (A.2) is calculated for every feasible portfolio, and the one with the lowest value is chosen. That is the new portfolio for state $\sigma$, which updates the policy. Last, the new policy is compared with the old one, determining if the algorithm has converged. If not, a new iteration round is started.
The discounted policy-iteration algorithm converges to the optimum after a finite number of iterations. This is guaranteed because the state space is finite, and for each state $\sigma$, the set of feasible maintenance action portfolios $Z_\sigma$ is finite as well (Puterman, 2014).

References


Barlow, R.E., Proschan, F., 1996. Mathematical theory of reliability. SIAM.


37
Leppinen et al.


Leppinen et al.


An Optimization Model for Determining Cost-Efficient Maintenance Policies for Multi-Component Systems with Economic and Structural Dependencies

Jussi Leppinen\textsuperscript{a,}\textsuperscript{*}, Antti Punkka\textsuperscript{b}, Tommi Ekholm\textsuperscript{c}, Ahti Salo\textsuperscript{a}

\textsuperscript{a}Systems Analysis Laboratory, Department of Mathematics and Systems Analysis, Aalto University, P.O. Box 11000, FI-00076 AALTO, FINLAND
\textsuperscript{b}SOK P.O. Box 1, FI-00088 S-RYHMA, FINLAND
\textsuperscript{c}Finnish Meteorological Institute P.O. Box 503, FI-00101 HELSINKI, FINLAND

\textsuperscript{*}Corresponding author

Email address: jussi.leppinen@aalto.fi (Jussi Leppinen)
• Technical structural dependencies affect the cost-efficiency of maintenance.
• Directed graphs can represent economic and structural dependencies of the system.
• The maintenance scheduling problem is modeled as a Markov Decision Process.
• Constraints arising from technical structural dependencies are included.
• Optimal maintenance policy complies with possible reliability requirements.
Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: