Constrained Multi-Aircraft Maintenance Scheduling Using Component Prognostics

A. Engelke

Delft University of Technology

TUDelft

Constrained Multi-Aircraft Maintenance Scheduling Using Component Prognostics

by

A. Engelke

Student number:	4204840	
Date:	July 23, 2019	
Thesis committee:	Dr. Ir. W.J.C. (Wim) Verhagen,	TU Delft, supervisor
	Dr. M.A. (Mihaela) Mitici,	TU Delft, supervisor
	Dr. Ir. E. (Erik-Jan) van Kampen,	TU Delft, external committee member
	F. (Floris) Freeman,	KLM, external committee member

TU Delft

Contents

List of Figures iii

List of Tables iv

List of Symbols v

List of Abbreviations vi

1 Introduction 1
- 1.1 Motivation and Relevance . . . 1
- 1.2 Research Objective and Questions . . . 1
- 1.3 Research Scope . . . 2
- 1.4 Structure of the Report . . . 2

I Scientific Paper 3

II Literature Study (previously graded under AE4020) 28

2 Prognostics 29
- 2.1 Prognostics Methods . . . 29
  - 2.1.1 On-line and Off-line PHM . . . 29
  - 2.1.2 Modeling Approaches . . . 29
- 2.2 Prognostics of Multi-Components . . . 33
- 2.3 Diagnostics and Prognostics State of the Art . . . 34

3 Airline Maintenance Schedules 36
- 3.1 General Airline Maintenance Methods . . . 36
  - 3.1.1 History of Airline Maintenance . . . 36
  - 3.1.2 Types of Airline Maintenance . . . 38
- 3.2 Current Research in Maintenance Scheduling . . . 41
  - 3.2.1 Line Maintenance . . . 41
  - 3.2.2 Base Maintenance . . . 42
- 3.3 Cost Factors in Airline Maintenance . . . 42
- 3.4 Current Status of Real-life Airline Maintenance Operations . . . 44

4 Maintenance Planning Optimization 46
- 4.1 Prognostics in Maintenance Planning Optimization . . . 46
- 4.2 Multi-Components in Maintenance Planning Optimization . . . 49
- 4.3 Optimization Algorithms . . . 52
  - 4.3.1 Genetic Algorithm . . . 52
  - 4.3.2 Markov Decision Process. . . . 52

III Elaborations of Thesis Work 54

5 Prognostics as an Input to the Model 55
- 5.1 Prognostics information obtained from the tool . . . 55

5.2 Failure Function Approximation 56
5.3 Implementation of the prognostics information in the MCTS 57

6 Monte-Carlo tree search Model Description 59
6.1 Tree Structure and Tree Characteristics 59
6.1.1 Layout of the Tree 59
6.1.2 Example Tree 61
6.1.3 Node Creation and Numbering 62
6.1.4 Available Maintenance Slots 62
6.1.5 Action and State 63
6.1.6 Prediction 64
6.1.7 Counter 65
6.1.8 Infeasibility Flag 66
6.2 Simulation 66
6.2.1 Simulated Action 66
6.2.2 Random Simulation 67
6.2.3 Award and Rollout Value Computations 67
6.3 Punishment Structure 69
6.3.1 Punishment Overview 69
6.3.2 Average Lifetime Punishments 69
6.3.3 Simulated Prognostics Punishment 69
6.3.4 Previous Prognostics Punishment 70
6.3.5 Expensive Maintenance Slot Punishment 70
6.4 MCTS Iteration Steps 71
6.4.1 Selection 71
6.4.2 Expansion 72
6.4.3 Simulation 72
6.4.4 Backpropagation 72

7 Additional Results 74

8 Discussion 76
8.1 Main Maintenance Scheduling Model 76
8.2 Probability Failure Distribution Prognostics 77
8.3 No Simulated Prognostics 77
8.4 Two Types of Aircraft 77

9 Verification and Validation 78
9.1 Verification 78
9.2 Validation 78

10 Conclusions and Recommendations 79

Bibliography 81

List of Figures

3.1 Schematic Drawing of the Failure Rate (Bathtub Curve) 37
3.2 Experiences Age-Reliability Relationships [52] 37
3.3 MSG-3 Level 1 Analysis [53] 40
3.4 MSG-3 Level 2 Analysis - Evident Failures [53] 41
3.5 Total Airline Cost Elements in 2014 [59] 43
3.6 Total Maintenance Cost [61] 44

5.1 Overview of the prognostics 56
5.2 Approximated Failure Function of the Bleed Air System 56

6.1 Time frame of the MCTS 59
6.2 MCTS after fourth iteration (with variables) 61
6.3 MCTS after fourth iteration (with sample values) 61
6.4 Sample node creation and numbering Part 1 62
6.5 Sample node creation and numbering Part 2 62
6.6 Sample node creation and numbering Part 3 62
6.7 Sample Possible Actions depending on State and maintenance opportunity 63
6.8 Sample counter computations 65
6.9 Sample Award Value Computations 68
6.10 Steps of one iteration of the MCTS approach 71

List of Tables

3.1 Overview of Common Types of Airline Maintenance 38

4.1 Overview of Research done with respect to Prognostics Implementation in Maintenance Planning Optimization 48
4.2 Overview of Research done with respect to Multi-Component Maintenance Planning Optimization 51
4.3 Overview of Optimization Approaches in Literature 53

5.1 Implementation of prognostics information 57
5.2 Types of predictions and their creation and usage in the MCTS 58

7.1 Scheduled Repair Actions 8 Aircraft of different types (scheduled together) 74
7.2 Scheduled Repair Actions 8 Aircraft of different types (scheduled individually) 74

List of Symbols

$\tilde{a}_{i,\tilde{j}}$	Simulated action for AC i on simulated layer $\tilde{j}$	[-]
a_{i,j,k_j}	Action to (not) repair AC i on node n_{j,k_j}	[-]
A_0	Award value of the root node on day 0	[-]
A_{j,k_j}	Award value of node n_{j,k_j}	[-]
$\overline{A}_{j,k_j}$	Average award up until node n_{j,k_j}	[-]
AO	Matrix storing the order of actions	[-]
c_w	weighting constant between exploitation and exploration of UCB1 formula	[-]
$C_{n,\tilde{j},i}$	Punishment n of AC i on simulated layer $\tilde{j}$	[-]
F_{j,k_j}	Infeasibility flag for node n_{j,k_j}	[-]
I	Amount of aircraft in the fleet	[-]
L_d	Number of days in the optimization / time window	[-]
L_j	j^{th} layer in the tree	[-]
L_p	Number of days in the past taken into account in the optimization (negative value)	[-]
$m_{i,j}$	Maintenance slots of AC i on day j	[-]
M_j	Maximum possible simultaneous repairs on day j	[-]
N	Amount of punishments considered	[-]
n_0	root node at day 0	[-]
n_{j,k_j}	k^{th} node on layer L_j	[-]
$P_{i,j}$	Prediction of failure for AC i on day j	[-]
P_p	Prediction matrix of the days prior to day 0	[-]
R_{j,k_j}	Random Rollout value of node n_{j,k_j}	[-]
R_m	Maximum possible random rollout value	[-]
S_{i,j,k_j}	State of AC i at node n_{j,k_j}	[-]
SM_i	Amount of specific maintenance slots for AC i in scheduling horizon	[-]
$t_{\bar{a}}$	Average life expectancy of a specific component	[-]
t_r	Time since last repair of a specific component of aircraft i	[-]
TM_i	Total amount of maintenance slots for AC i in scheduling horizon	[-]
$UCB1_{j,k_j}$	UCB1 value of node n_{j,k_j}	[-]
x_0	Counter of how often the root node on day 0 was included in a simulation	[-]
x_{j,k_j}	Counter of how often node n_{j,k_j} was included in a simulation	[-]

List of Abbreviations

CBM	Condition-based maintenance
GM	Generic maintenance slot
LCC	Low-cost carrier
MCTS	Monte-Carlo tree search
NPV	Negative Predictive Value
PHM	Prognostics & Health Management
PPV	Positive Predictive Value
RUL	Remaining useful lifetime
SM	Specific maintenance slot
TSP	Traveling salesman problem

1
Introduction

1.1. Motivation and Relevance

Commercial airlines underwent large changes during the last couple of decades. A wave of aircraft with growing passenger capacities and flight distances formed the airline network schedules around the turn of the century. The main legacy carriers dominated the market with service-oriented strategies. A crucial change in the industry, however, was the market growth of low-cost carriers (LCC) [1]. Their strategy to offer flights at minimum costs forced legacy carriers to reduce their cost as well to stay competitive. The first response of airlines and therefore also of researchers in that field was to improve their networks, routes, general schedules, and flight operations. A lot of work within that area has been done and incorporated into actual airline operations. It was and still is however difficult to indeed improve maintenance schedules for a long time. Which was mainly due to the underdeveloped knowledge in the field of failure diagnostics and prognostics. The development of sensors and computational algorithms to track, measure and detect degradation processes, however, allowed advancements in the field of prognostics and health management (PHM) [2]. Currently, many studies are working on either developing prognostics or on developing maintenance planning optimization. However, only a few research studies are performed in linking those two aspects into one maintenance schedule.

1.2. Research Objective and Questions

The main objective of this research is to develop a component maintenance schedule for multiple aircraft. It is furthermore important to include PHM information of that system and to implement a real-life environment. To reach this objective, some smaller goals need to be reached.

- Select existing PHM classification algorithm to use as an input to a maintenance scheduling approach
- Translate airline maintenance schedules into maintenance slots and other constraints
- Develop a model to schedule maintenance actions based on failure prediction and available maintenance slots

Concluding from the above-mentioned research aim and objective and the state of the art concerning applying prognostics to multi-unit component maintenance schedules, the following main research question can be formulated.

How can component prognostics estimates be utilized in multi-aircraft maintenance scheduling to minimize wasted lifetime while avoiding unscheduled maintenance?

To answers this overall question it is crucial to analyze a variety of aspects which are formulated in the research sub-questions below.

1. How can component prognostics and maintenance slots be utilized in a maintenance schedule?
   (a) How can it be determined which prognostics should be used?
   (b) How can classification prognostics be used in a scheduling approach?
   (c) How can fixed and flexible maintenance slots be incorporated in the optimization?
2. How will the model of an optimal maintenance strategy be built up?
   (a) Which approach should be chosen for the scheduling assignment?
   (b) Which factors should be taken into account for scheduling aircraft?
   (c) How can a maintenance scheduling assignment be formulated in a mathematical model?

1.3. Research Scope

The research develops a repair schedule for 20 Boeing 777 of a European airline. To do so the research implements classification prognostics developed for the Bleed Air System of a wide-body fleet of the same European Airline and the available maintenance slots of these 20 investigated aircraft. This means that the study investigates multiple units of the chosen component. It is however out of the scope to develop the prognostics and therefore these inputs are artificially generated. The optimization is further constrained in a way that only existing maintenance opportunities are part of the scope of the scheduling process instead of optimizing the opportunities themselves.

1.4. Structure of the Report

The report is structured as follows. At first, a scientific paper is presented in Part I to concisely elaborate on the research done. The research done is supported by a literature study in Part II. It presents the state of the art of prognostics in Chapter 2, airline maintenance scheduling in Chapter 3 and maintenance planning optimization in Chapter 4. Finally, Part III further elaborates on the research by explaining in more detail the prognostics used for the maintenance scheduling model in Chapter 5. After which the full model is presented in more detail in Chapter 6. Then in Chapter 7 additional results to the ones presented in Part I are shown. The discussion of the full set of results and the additional results can then be found in Chapter 8. Verification and Validation of the maintenance scheduling algorithm can be found in Chapter 9. Finally, the conclusions and recommendations of the research can be found in Chapter 10.

I

Scientific Paper

Constrained multi-aircraft maintenance scheduling using component prognostics

Author: A. Engelke
Supervisors: M.A. Mitici and W.J.C. Verhagen
Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

Abstract

In recent years, airlines have increasingly developed the ability to monitor the condition of aircraft components by means of sensors. In turn, aircraft maintenance aims to use this sensor data to predict component failures. However, the challenge remains to make use of these prognostics to generate appropriate maintenance schedules. In this paper, we develop a Monte-Carlo tree search to schedule maintenance tasks based on component prognostics and available maintenance slots. This approach is used to create a maintenance policy for multiple aircraft which specifies which aircraft are allocated for maintenance and on which days. The results show that the scheduling of the maintenance tasks is robust and able to accommodate the maintenance scheduling of smaller airline fleet sizes. Overall, our results support the integration of aircraft component prognostics in aircraft maintenance scheduling.

Keywords: Multi-Aircraft, Prognostics, Maintenance Scheduling, Monte-Carlo Tree Search

1. Introduction

The airline industry underwent large changes during the last decades and especially the market growth of low-cost carriers (LCC) [1] forced legacy carriers to reduce costs of their operations and therefore also of their maintenance activities.

An important development in the maintenance field is the development of failure prognostics. The improvements of sensors and computational algorithms to track, measure and detect degradation processes allowed advancements in the field of prognostics and health management [2] in order to predict failure. These advancements allow for new and improved maintenance schedules in the industry. Condition-based maintenance (CBM) policies propose a framework to incorporate prognostic advancements into maintenance scheduling to achieve higher system availability and cost reduction. A US study of 2003 predicted a 35 billion dollar per year cost reduction in the US if CBM would be fully utilized to minimize unexpected downtimes [3]. Unfortunately, a more recent study found that companies actually observe a large difference between the potential of prognostics and actually achieved benefits [4]. Apparently, more research is needed to find solutions which offer a complete and easy to use application incorporating a variety of CBM aspects, which furthermore are linked to real-life airline operations. The main challenge it to draw operational conclusions from the newly gained prognostics information. Meaning that operators need to answer the question 'When to repair an aircraft when a remaining useful lifetime (RUL) is presented by a prognostics tool?'.

In this paper, we address this challenge of scheduling a repair action based on known prognostics information about a specific component. The approach is constrained by available maintenance slots and opts to optimize not one unit but multiple aircraft of an airline fleet.

The remainder of this paper is structured as follows. In Section 2 we elaborate on the current state of the art of research conducted in the field of maintenance scheduling having component prognostics as input. We also review current research efforts using Monte-Carlo tree search (MCTS) models. In Section 3 we present a Monte-Carlo tree search approach to schedule maintenance tasks using component prognostics. We illustrate our results by means of a case study in Section 4. In Section 5 we propose two additional case studies and present the results. In Section 6 we provide a discussion of the results. Section 7 provides conclusions and recommendations.

2. Prior Work

In recent years, a variety of studies worked on topics related to the presented scheduling approach. At first we present the research done in the field of maintenance planning optimization, followed by other applications of the chosen modeling approach (MCTS).

2.1. Maintenance Planning

A number of studies have been performed to include newly gained prognostics information in a maintenance schedule, some within a contextual framework like wind farms, railways or aircraft operations and others with a more research related focus.

One important difference consists of which type of prognostics are considered in the planning approach. Most commonly models include Remaining Useful Lifetime (RUL) [5, 6, 7] or failure probability prognostics as an input [8, 9]. All studies, however, assume that the RUL information or failure probabilities are correct and therefore the accuracy of these values is not taken into account.

The simplified implementation of prognostics information is furthermore often limited by applying thresholds and triggers of minimum RULs or minimum failure probability before the planning algorithm is activated [6, 9, 7]. Another difference between different approaches is the objective function applied to the problem. It varies from maximizing revenue [5] to minimizing (maintenance) cost [8, 7], minimizing risk [6, 9] and minimizing unused maintenance slots [10]. Due to the fact that many studies consider no or a very limited operational context one of their main assumptions also is that immediate repair is possible at all times [8, 6, 7].

Other studies focus on finding solutions in a multi-unit planning framework to schedule multiple systems simultaneously and considering their constraints on each other. Besides looking at cost often other objectives are also considered. It is possible to minimize not just cost but also the multi-unit unavailability [11], maximize productivity of all units [12] or to minimize labor, repair and parts costs [13].

2.2. Monte-Carlo Tree Search Application

The Monte-Carlo tree search is a reinforcement-learning algorithm which first appeared in 2006 [14, 15] and enabled Google DeepMind to develop AlphaGo, an artificial intelligent Go-playing algorithm, and to defeat the human world champion Go player [16]. This approach was chosen since the Monte-Carlo tree search is able to present a solution without a pre-defined objective function and without large amounts of data input. Which is especially useful in the situation at hand with simplified prognostics input.

The most common applications of the MCTS are in the field of game theory and game solving programes like AlphaGo. Applications can be found for the games Go [17, 18], Othello [19, 20] and many others. But since the Monte-Carlo Tree Search does not require detailed information about the context it is also highly suitable for non-game applications. The usability ranges from solving traveling salesman problems (TSP) [21, 22] and a variety of scheduling problems [23, 24, 25, 26].

3. Model Description and Formulation

In this section, we propose a Monte-Carlo tree search to schedule maintenance tasks of multiple aircraft based on component prognostics. We first introduce the generic concept of the Monte-Carlo tree search. Further, we introduce the component prognostics and show how they are taken into account into the maintenance scheduling tree search. Then, we define an appropriate search tree to represent the aircraft maintenance scheduling. Lastly, we define the value function of the Monte-Carlo tree search for the aircraft maintenance schedule and elaborate on the exploration and exploitation of this Monte-Carlo tree search.

3.1. Generic Monte-Carlo tree search concept

The MCTS [14, 15] is a reinforcement learning approach which iteratively samples from a given node down the tree. Each iteration consists of four steps: selection, expansion, simulation ,and back propagation (see also Figure 1). Each iteration starts at the root moving down the tree and selecting a not yet expanded node according to a selection criterion as the Upper Confidence Bound (UCB1) (selection phase). Then all possible child nodes are initialized below the selected node (expansion phase) after which the algorithm simulates a possible path until a terminal state is reached (simulation phase). The selected path and the simulated path are then evaluated according to a value function and the evaluation and a counter of how often a node has been visited are backpropagated to update the statistics of all nodes in the selected path (backpropagation phase).

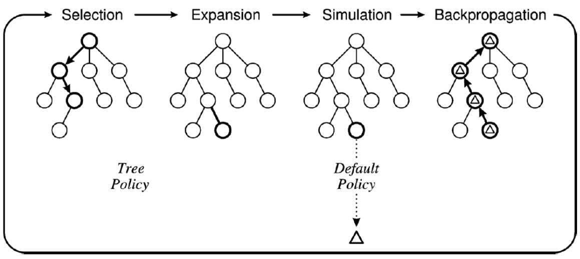


Figure 1: Steps of the general MCTS approach [27]

3.2. Component prognostics as input to the maintenance scheduling Monte-Carlo tree search

We define the aircraft maintenance scheduling as a Monte-Carlo tree search. One input of the tree is the prognostic of the aircraft component failure $P_{i,j}$ for aircraft i for a time horizon of L_d days in the future, where $P_{i,j} \in \{0, 1\}$ is a classification prognostic that indicates:

$$P_{i,j} = \begin{cases} 1, & \text{is the prediction at day } j \text{ that aircraft } i \text{ is subject to component failure in the next } L_d \text{ days,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We also consider previous prognostics up to L_p days in advance from the first day of the scheduling horizon, where $P_{i,-j} \in \{1, 0\}, j \in \{1, 2, \ldots, L_p\}$, with

$$P_{i,-j} = \begin{cases} 1, & \text{is the prediction at day } -j \text{ that aircraft } i \text{ is subject to component failure in next } L_d \text{ days,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The $P_{i,-j}$ prognostics are assumed to be known at the beginning of the scheduling horizon of L_d days. The $P_{i,j}$ prognostics are unknown at the moment of decision making for maintenance scheduling. These prognostics are used in the Monte-Carlo tree search to generate random scenarios associated with the failure of a component in the next L_d days. In a similar fashion, in a Go game [16] these prognostics are seen as possible, future moves to reach a terminal state.

We consider that these classification prognostics have an accuracy specified in the form of a prognostic negative predictive value (NPV) and a positive predictive value (PPV) [28], where:

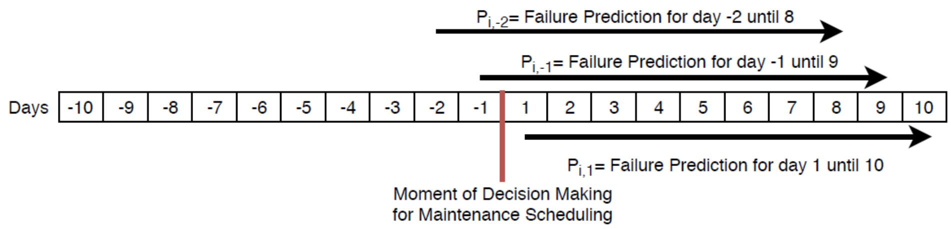
$$NPV = \frac{TN}{TN + FN} \tag{3}$$

$$PPV = \frac{TP}{TP + FP}, \tag{4}$$

where TN is the number of true negatives associated with the prognostic, FN is the number of false negatives associated with the prognostics, TP is the number of true positives associated with the prognostic and FP is the number of true negatives associated with the prognostics.

Figure 2 illustrates the concept of component prognostics where $L_p = 10$ and $L_d = 10$ days before and after the moment of decision making, respectively. For each of the past 10 days, we consider known classification component prognostics $P_{i,-j}$, $1 \leq j \leq 10$. Each of these prognostics indicates whether this component will fail or not in the next 10 days with an accuracy of NPV and PPV. For each of the next $L_d = 10$ days, we simulate prognostics for the component failure. Similarly, each of these simulated prognostics indicates whether this component will fail or not in the next $L_d = 10$ days with an accuracy of NPV and PPV. At the moment of decision making, these prognostics are among the main inputs to decide whether the component is scheduled for maintenance or not in the next $L_d = 10$ days.

Figure 2: Example of prognostics considering 10 days prior and 10 days after initialisation, where at initialisation a decision about maintenance scheduling is made



3.3. *Monte-Carlo tree search for the aircraft maintenance scheduling*

We represent the aircraft maintenance scheduling by means of a tree. We consider a fleet of I aircraft, $|I| \geq 1$. We consider a time horizon of L_d days, $L_d > 0$, over which a prognostic is made on the failure of a component for each aircraft. Based on these prognostics, we also consider a scheduling horizon of L_d days, when an aircraft can be subject to maintenance or not. We define a search tree with L_d layers, where each layer corresponds to a day in the scheduling horizon. Further, each layer L_j, $1 \leq j \leq L_d$, has k_j nodes, $1 \leq k_j \leq n^j_{child}$. Each node k_j has n_{child} maximum amount of child nodes. Thus, layer L_j has a maximum amount of n^j_{child} nodes.

For each layer L_j, which is the j^{th} layer of the tree, we define the following information for all aircraft $i \in I$ and day j in the scheduling horizon $1 \leq j \leq L_d$:

$$L_j : (m_{1,j}, m_{2,j}, ..., m_{i,j}; M_j; P_{1,j}, P_{2,j}, ..., P_{i,j}), \tag{5}$$

where $m_{i,j}$ is the possibility for aircraft i to make use of an available maintenance slot in day j. We consider two types of maintenance slots: aircraft tail-specific (SM) slot, generic (GM) slot. Thus, $m_{i,j} \in \{SM, GM, 0\}$, where $m_{i,j} = SM$ means that aircraft i has the possibility to be maintained during a tail-number specific maintenance slot in day j, $m_{i,j} = GM$ means that aircraft i has the possibility to be maintained during a generic maintenance slot in day j and $m_{i,j} = 0$ means that there is no slot available to maintain aircraft i in day j. We also consider $M_j, M_j > 0$ to be the maximum amount of aircraft that can simultaneously undergo maintenance on day j.

Each tree node n_{j,k_j}, which is the k^{th} node in layer L_j, is defined as follows, for all $i \in I$, $1 \leq j \leq L_d$ and $1 \leq k_j \leq n^j_{child}$:

$$n_{j,k_j} : (a_{1,j,k_j}, a_{2,j,k_j}, ..., a_{i,j,k_j}; S_{1,j,k_j}, S_{2,j,k_j}, ..., S_{i,j,k_j}), \tag{6}$$

where $a_{i,j,k_j} \in \{0, 1\}$ is the maintenance scheduling action taken for aircraft i at node n_{j,k_j} (at day j), with

$$a_{i,j,k_j} = \begin{cases} 1, & \text{if aircraft } i \text{ scheduled for maintenance at node } n_{j,k_j} \text{ (maintenance at day } j) , \\ 0, & \text{if aircraft } i \text{ not scheduled for maintenance at node } n_{j,k_j} \text{ (no maintenance at day } j) . \end{cases} \tag{7}$$

Also, $S_{i,j,k_j} \in \{0, 1\}$ defines the maintenance state of aircraft i at node n_{j,k_j}, where

$$S_{i,j,k_j} = \begin{cases} 1, & \text{if at node } n_{j,k_j}, \text{ aircraft } i \text{ has undergone maintenance between day 0 and day } j \\ 0 & \text{if at node } n_{j,k_j}, \text{ aircraft } i \text{ has not undergone maintenance between day 0 and day } j \end{cases} \tag{8}$$

We also consider a root node n_0 which corresponds to the initialization of the tree search, and, thus, does not have an associated action or aircraft state.

We define a feasible maintenance actions a_{i,j,k_j} based on the state $S_{i,j-1,k_{j-1}}$ of the previous (parent) node $n_{j-1,k_{j-1}}$ and the maintenance opportunity $m_{i,j}$ at layer L_j, as follows

$$a_{i,j,k_j}\left(S_{i,j-1,k_{j-1}}, m_{i,j}\right) = \begin{cases} 0 \text{ or } 1 & \text{if } S_{i,j-1,k_{j-1}} = 0 \ \textbf{ and } \ m_{i,j} \neq 0 \\ 0 & \text{if } S_{i,j-1,k_{j-1}} = 1 \\ 0 & \text{if } m_{i,j} = 0. \end{cases} \tag{9}$$

From eq. (9) an action to maintain the component of aircraft i, i.e., $a_{i,j,k_j} = 1$ at most one time during the scheduling horizon of L_d days.

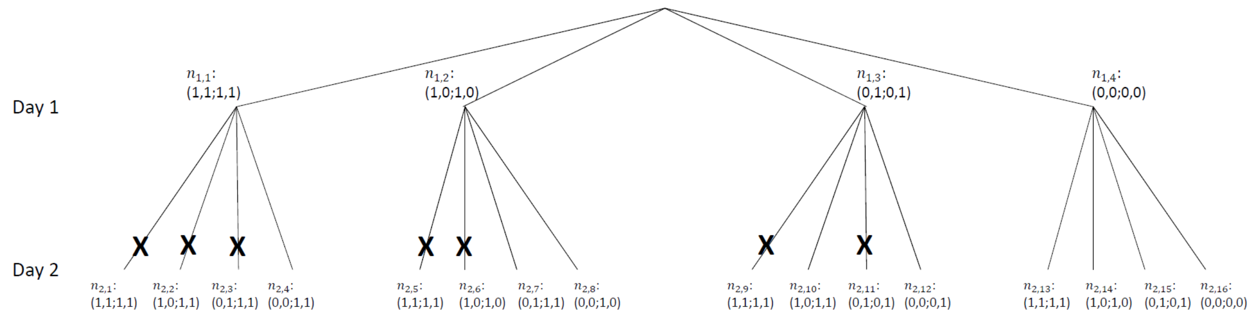
Also, for a given layer L_j, $1 \leq j \leq L_d$, and a given node n_{j,k_j}, $1 \leq k_j \leq n^j_{child}$, we also ensure that the amount of aircraft that are scheduled for maintenance in day j does not exceed the maximum number of simultaneous maintenance actions M_j, i.e.,

$$\sum_{i=1}^{I} a_{i,j,k_j} \leq M_j \tag{10}$$

Lastly, we update the state S_{i,j,k_j} of aircraft i at node n_{j,k_j} depending on the state of the previous visited node $n_{j-1,k_{j-1}}$ and the action a_{i,j,k_j} as follows:

$$S_{i,j,k_j}\big(S_{i,j-1,k_{j-1}}, a_{i,j,k_j}\big) = min\big\{S_{i,j-1,k_{j-1}} + a_{i,j,k_j}, 1\big\}. \tag{11}$$

Figure 3: Example setup of the first two layers of a tree with I=2, where X marks an infeasible node



3.4. Monte-Carlo Tree Search Algorithm

In this section, we show how the maintenance scheduling Monte-Carlo tree search is explored and exploited. In doing so, we make use of punishments, which we introduce at the beginning of this section. In the second part of this section we then show how the four generic steps of the Monte-Carlo tree search algorithm are implemented in our maintenance scheduling approach, this includes that we show how simulation of the Monte-Carlo tree search is conducted.

3.4.1. Punishments in the Monte Carlo tree search

We define 5 punishments to decide whether to schedule an aircraft for maintenance or not. These punishments are evaluated at the level of each node n_{j,k_j} of the selected path in the Monte Carlo tree search. In order to compute the punishments, we define a set L of all nodes n_{j,k_j} along the selected path. The set, therefore, consists of L_d elements.

$$L = \{n_{1,k_j}, n_{2,k_j}, ..., n_{L_d,k_j}\} \tag{12}$$

The punishment values $C_{n,j,i,l}$ are defined on each layer j, $1 \leq j \leq L_d$, for each punishment n, $1 \leq n \leq 5$, for each aircraft i, $i \in I$ and for each node $k_j \in L$.

Average Lifetime Punishments

Equation (13) shows a punishment which sanctions a choice to repair aircraft i if the time since last repair t_r of that aircraft i is lower than the mean time between repair $t_{\bar{a}}$. Therefore the likelihood is reduced that an aircraft, which has been repaired a comparatively short time ago, is scheduled for a repair.

$$C_{1,j,i,l} = \begin{cases} 1 - \frac{t_r}{t_{\bar{a}}} & \text{if } t_r < t_{\bar{a}} \text{ and } S_{i,j,k_j} = 1 \\ 1 - \frac{t_r}{t_{\bar{a}}} & \text{if } t_r < t_{\bar{a}} \text{ and } a_{i,j,k_j} = 1 \\ 0 & \text{if } t_r \geq t_{\bar{a}} \\ 0 & \text{if } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \end{cases} \tag{13}$$

If an aircraft i with a time since last repair t_r larger than the mean time between repair $t_{\bar{a}}$ is not scheduled it receives a punishment according to equation (14). Similarly to punishment $C_{1,j,i,l}$, it is, therefore, less likely to not schedule an aircraft if it has not been repaired for a comparatively long time.

$$C_{2,j,i,l} = \begin{cases} 1 - \frac{t_{\bar{a}}}{t_r} & \text{if } t_r > t_{\bar{a}} \text{ and } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \\ 0 & \text{if } t_r \leq t_{\bar{a}} \\ 0 & \text{if } S_{i,j,k_j} = 1 \\ 0 & \text{if } a_{i,j,k_j} = 1 \end{cases} \tag{14}$$

Simulated prognostics punishment

An action a_{i,j,k_j} is punished according to equation (15), depending on whether or not it matches the predictions $P_{i,j}$ simulated for that day. An aircraft is, therefore, less likely to be scheduled if there are few failure predictions and more likely to be scheduled if there are many failure predictions.

$$C_{3,j,i,l} = \begin{cases} NPV & \text{if } P_{i,j} = 0 \text{ and } a_{i,j,k_j} = 1 \\ (1 - NPV) & \text{if } P_{i,j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } NPV \geq 0.5 \\ 0 & \text{if } P_{i,j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } NPV < 0.5 \\ (1 - PPV) & \text{if } P_{i,j} = 1 \text{ and } a_{i,j,k_j} = 1 \text{ and } PPV \geq 0.5 \\ 0 & \text{if } P_{i,j} = 1 \text{ and } a_{i,j,k_j} = 1 \text{ and } PPV < 0.5 \\ PPV & \text{if } P_{i,j} = 1 \text{ and } a_{i,j,k_j} = 0 \end{cases} \tag{15}$$

Previous prognostics punishment

The predictions are made for the next ten days, meaning that a prediction is made on a specific day whether or not the component will fail in the upcoming ten days. Equation (16) punishes a sequence of ten days if the number of repairs in those ten days do not match the prediction made for that 10 day period. Therefore it will be less likely that an aircraft is scheduled if the failure predictions in the past were mostly negative and similarly it will be more likely to be scheduled if many of the previous predictions were positive.

$$C_{4,j,i,l} = \begin{cases} PPV & \text{if } P_{i,j-10} = 1 \text{ and } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \\ (1 - PPV) & \text{if } P_{i,j-10} = 1 \text{ and } (S_{i,j,k_j} = 1 \text{ or } a_{i,j,k_j} = 1) \text{ and } PPV \geq 0.5 \\ 0 & \text{if } P_{i,j-10} = 1 \text{ and } (S_{i,j,k_j} = 1 \text{ or } a_{i,j,k_j} = 1) \text{ and } PPV < 0.5 \\ NPV & \text{if } P_{i,j-10} = 0 \text{ and } (S_{i,j,k_j} = 1 \text{ or } a_{i,j,k_j} = 1 \\ (1 - NPV) & \text{if } P_{i,j-10} = 0 \text{ and } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } NPV \geq 0.5 \\ 0 & \text{if } P_{i,j-10} = 0 \text{ and } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } NPV < 0.5 \end{cases} \tag{16}$$

Expensive maintenance slot punishment

Not all maintenance slots are identical and therefore if a repair is scheduled during one of the more expensive opportunities ($m_{i,j} = GM$) that action is punished according to equations (17). Meaning that an aircraft is less likely to be scheduled during a general maintenance slot instead of a specific maintenance slot.

$$C_{5,j,i,l} = \begin{cases} \frac{SM_i}{TM_i} & \text{if } a_{i,j,k_j} = 1 \text{ and } m_{i,j} = GM \\ 0 & \text{if } a_{i,j,k_j} = 0 \\ 0 & \text{if } m_{i,j} = SM \end{cases} \tag{17}$$

3.4.2. Monte-Carlo Tree Search Steps

As shown in Figure 1 the Monte-Carlo tree search algorithm consists of four steps. In order to apply them we define a rollout value $R_{j,k_j} \in \mathbb{R}^+$, an award value $A_{j,k_j} \in \mathbb{R}^+$, a UCB1 value $UCB1_{j,k_j} \in \mathbb{R}^+$, a counter x_{j,k_j} of node n_{j,k_j} and a counter x_0 of the root. All node characteristics are initialized equal to zero during initialization of node n_{j,k_j}. How we implement these node characteristics in the MCTS steps can be seen below.

A. Selection

The Monte-Carlo tree search tries to balance its iterations between exploration and exploitation to cover the nodes with the highest potential with the least computational effort. Meaning that it selects nodes n_{j,k_j} with high award values A_{j,k_j} and with few visits x_{j,k_j}. The Upper Confidence Bounds (UCB1) algorithm can be used to select a node [29]. A trade off is made between exploitation (*Term:* $\overline{A}_{j,k}$) and exploration (*Term:* $\sqrt{\frac{ln(x_0)}{x_{j,k}}}$, where x_0 is the counter of the root node). A weighting constant c_w is used to balance those aspects, where $c_w > 0$ and from theory it has a default value of $c_w = \sqrt{2}$ [29]. The UCB1 value $UCB1_{j,k_j}$ is therefore defined for each node n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$ and $1 \leq j \leq L_d$:

$$UCB1_{j,k_j} = \overline{A}_{j,k_j} + c_w\sqrt{\frac{ln(x_0)}{x_{j,k_j}}} \tag{18}$$

The average award value of node n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$ and $1 \leq j \leq L_d$, can be calculated by dividing the award value A_{j,k_j} by the amount of visits x_{j,k_j} of node n_{j,k_j}.

$$\overline{A}_{j,k_j} = \frac{A_{j,k_j}}{x_{j,k_j}} \tag{19}$$

B. Expansion

If the selected node n_{j,k_j} has been visited and simulated once before, $x_{j,k_j} = 1$, it needs to be expanded. This means that all n_{child} possible child nodes are added into the tree below node n_{j,k_j}. We define all node statistics (action a_{j,k_j}, state S_{j,k_j}, award value A_{j,k_j}, rollout value R_{j,k_j}, UCB1 value $UCB1_{j,k_j}$ and counter x_{j,k_j}) for the new child nodes during initialization.

In case the node n_{j,k_j} has not been visited before, $x_{j,k_j} = 0$, this step is skipped and the node is directly rolled out in a random simulation as explained in the step *C. Simulation*.

C. Simulation

A simulation can start at a feasible node n_{j,k_j} which has not been rolled out before ($x_{j,k_j} = 0$). We define this starting layer of the simulation as j', where $1 \leq j' \leq L_d$. When a such a node is found it can be rolled out in a random simulation which means that for each layer $j' \leq j \leq L_d$, and for each aircraft i, $i \in I$, a random simulated action a_{i,j,k_j} is compared with the prediction $P_{i,j}$ made for that layer. The comparison is determined and documented in the punishments $C_{n,j,i,l}$, which will then be used in the rollout value R_{j,k_j} and award value A_{j,k_j}.

Each node n_{j,k_j} is evaluated based on rewards shown during random rollout simulations. During these simulations, a random combination of actions and punishments are simulated. These simulated actions a_{i,j,k_j} are not a node property but an action which is randomly chosen in the simulation process for each layer j, $j' \leq j \leq L_d$ and for each aircraft i, $i \in I$.

$$a_{i,j,k_j} \in \{0,1\} \tag{20}$$

$$a_{i,j,k_j} = \begin{cases} 1 & \text{if the random rollout chooses to repair AC } i \text{ on day } j \text{ during simulation} \\ 0 & \text{if the random rollout chooses not to repair AC } i \text{ on day } j \text{ during simulation} \end{cases} \tag{21}$$

Which simulated action a_{i,j,k_j} is chosen in the simulation for aircraft i on layer j is constrained by same aspects as defined in equations (9) and (10).

Each node n_{j,k_j} is furthermore equipped with an award value A_{j,k_j} and a random rollout value R_{j,k_j}. The random rollout value is computed using a random simulation of the next $L_d - j$ days and it is restricted by a maximum possible rollout value R_m. The award value A_{j,k_j} of node n_{j,k_j} is equal to the rollout value R_{j,k_j} of node n_{j,k_j} and the sum of the award values of the child nodes of node n_{j,k_j} as can be seen in equation (22).

$$A_{j,k_j} = \begin{cases} 0 & \text{if } x_{j,k_j} = 0 \\ R_{j,k_j} & \text{if } x_{j,k_j} = 1 \\ R_{j,k_j} + \sum_{u=n_{child}\cdot(k-1)+1}^{k\cdot n_{child}} A_{j+1,u} & \text{if } x_{j,k_j} > 1 \end{cases} \tag{22}$$

$$R_m = N \cdot (L_d - 1) \cdot I \tag{23}$$

$$R_{j,k_j} = \begin{cases} 0 & \text{if } x_{j,k_j} = 0 \\ \frac{R_m - \sum_{n=1}^{N}\sum_{j=j}^{L_d}\sum_{i=1}^{I} C_{n,j,i,l}}{R_m} & \text{if } x_{j,k_j} > 0 \end{cases} \tag{24}$$

The random rollout value can be computed using a maximum possible rollout value R_m subtracted by the sum of all punishments n, $1 \leq n \leq N$, on node n_{j,k_j}, with $1 \leq k_j \leq n_{child}$ and $1 \leq j \leq L_d$, per aircraft i, $i \in I$.

The maximum possible random rollout value R_m is dependent on the number of punishments N, amount of days into the past L_d and the amount of aircraft I.

Therefore a rollout value R_{j,k_j} close to R_m shows a node with positive potential and few punishments in the random simulation.

D. Backpropagation

Once a rollout value R_{j,k_j} is found for the leaf node n_{j,k_j} the award value A_{j,k_j} and the counter x_{j,k_j} of that node are updated. We then update the award values and counters along the path up until node n_{j,k_j} accordingly.

4. Results

In this section, we describe the results of the Monte-Carlo tree search algorithm as maintenance scheduling model. We first present the implemented input data of the time since last repair, available maintenance slots and classification component prognostics and how these are obtained. After which we show and elaborate on the numerical results of the main model.

4.1. Data used in case study

We consider a fleet of 20 Boeing 777 aircraft of a European airline, where we consider their time since last repair, available maintenance slots and classification prognostics. In order to compare the results a standard input data set is used.

Time Since Last Repair

The time since last repair, t_r, of aircraft i, is randomly sampled based on the mean time between repairs, $t_{\bar{a}}$, of the component analyzed in the used component prognostics (see equation (25)). The data of all 20 aircraft are shown in Table 2 directly linked to the results of the maintenance scheduling of 20 aircraft.

$$t_r \sim U(0, 2t_{\bar{a}}) \tag{25}$$

Available maintenance slots

The information about available maintenance slots $m_{i,j}$ is based on planned maintenance slots per aircraft and generally available maintenance slots for a specific type provided by the European airline used for this case study. A maintenance slot somewhere on a specific day is considered an opportunity for that day without evaluating whether the planned maintenance leaves enough time for another maintenance action.

The maintenance slots considered in this case study can be seen in Table 2 directly linked to the results of the maintenance scheduling model.

We, furthermore, consider a maximum amount of simultaneous repairs M_j of one aircraft per day for all days j.

Prognostics for component failure

The component prognostics $P_{i,j}$ in real life would only be known from the previous days and not of the upcoming days. As part of the Monte-Carlo tree search random simulation, those are however required in a form of a potential possible situation. The previous predictions are generated upon initialization by a random generator based on failure probabilities related to the time since last repair t_r of aircraft i. Whereas the component prognostics of L_d days in the future are randomly generated during each simulation iteration. The classification prognostics used as an input in this case study does not provide any failure probability information and therefore an approximation needs to be found if one wishes to include a certain increase in failure probabilities. Since 94% of all components show a constant hazard function [30] a failure distribution which leads to a constant hazard function is most likely to be correct. The exponential distribution is the only continuous distribution with a constant hazard function and therefore an exponential function is assumed for future steps. The rate parameter λ describes the average failure rate. The component (bleed air system) analyzed in the component prognostics used as an input in this case study, on average fails once in 354 days. This mean time between repairs will furthermore also be used in the simulation in the form of $t_{\bar{a}} = 354$. The failure function can then be calculated as shown in equation (26).

$$F(t_r) = 1 - exp(-\frac{1}{354} \cdot t_r) \tag{26}$$

Knowing the time since last repair it is, therefore, possible to approximate the failure probability at a specific day. Which we then use as a bias when randomly creating the component prognostics based on the failure probability $F(t_r)$ (see equation (27)).

$$P_{i,j} = \begin{cases} 1 & \text{, with probability } 1 - exp(-\frac{1}{354} \cdot t_r) \\ 0 & \text{, otherwise} \end{cases} \tag{27}$$

The result of this biased generation of the days prior to initialization is presented in Table 1. Whereas the predictions used for days in the future are created following the same rules during each simulation.

Table 1: Classification Prognostics Input

AC \ Day	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
1	0	0	0	0	0	1	1	0	1	0
2	0	1	0	1	1	0	0	0	0	1
3	0	1	0	1	0	0	0	0	0	0
4	1	1	1	1	0	1	1	1	1	1
5	1	0	1	1	1	0	1	1	1	0
6	1	0	0	1	0	0	0	1	0	1
7	0	1	1	0	1	1	1	1	0	1
8	0	1	0	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	0	1	1
10	1	1	0	1	1	1	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0
12	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	1	0
14	1	1	0	1	1	1	0	0	1	0
15	0	1	1	0	1	0	1	1	1	0
16	1	1	1	1	1	1	0	1	0	1
17	1	1	1	0	1	0	1	1	1	1
18	0	0	0	0	0	0	1	1	0	1
19	0	1	1	0	0	0	1	0	1	1
20	0	0	0	0	1	0	0	1	0	0

In this paper, we consider classification component prognostics developed for the bleed air system of the fleet of a European airline, where we assume the following NPV and PPV values: $NPV = 0.86$ and $PPV = 0.31$. Meaning that only 31% of the yes predictions correctly predict a failure in the upcoming L_d days and that 86% of the no predictions correctly predict that no failure occurs in the next L_d days. We considered these NPV and PPV values in Section 3.4 when presenting the punishment structure of the award values.

4.2. Numerical Results: Maintenance Scheduling of aircraft fleet

The proposed application of our Monte-Carlo tree search algorithm for maintenance scheduling is evaluated in form of its main model description. The main model is used to schedule 20 aircraft and the results are then analyzed with respect to their correctness, robustness, computational time and sensitivity to varying input.

Scheduling 20 Aircraft

Table 2 presents the scheduled repair actions of a fleet of 20 aircraft. The table also shows the time since last repair t_r of each aircraft and the maintenance opportunities $m_{i,j}$ (at which day and what type of opportunity), where SM are specific maintenance slots available for that aircraft and GM are general maintenance slots available for all aircraft of that type.

Table 2: Scheduled Repair Actions 20 Aircraft

AC 1			AC 2			AC 3			AC 4			AC 5		
t_r = 178			t_r = 245			t_r = 233			t_r = 657			t_r = 536		
m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?
1	SM	-	8	SM	-				1	SM	-	8	SM	-
4	SM	-							4	SM	✓			
7	SM	✓							5	GM	-			
8	GM	-							7	SM	-			
									8	SM	-			

AC 6			AC 7			AC 8			AC 9			AC 10		
t_r = 205			t_r = 430			t_r = 543			t_r = 600			t_r = 639		
m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?
			2	GM	✓				1	SM	✓	8	SM	-
			8	SM	-				4	GM	-			
									7	SM	-			

AC 11			AC 12			AC 13			AC 14			AC 15		
t_r = 422			t_r = 49			t_r = 155			t_r = 616			t_r = 294		
m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?
1	SM	-	1	GM	-	8	SM	✓				8	SM	-
4	SM	-	4	SM	-	9	GM	-						
7	SM	-	7	SM	-									

AC 16			AC 17			AC 18			AC 19			AC 20		
t_r = 469			t_r = 555			t_r = 176			t_r = 393			t_r = 163		
m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?	m_ij Day	Type	Repair?
5	SM	✓	3	SM	✓	1	SM	-	1	SM	-	1	SM	-
8	SM	-	5	SM	-	3	GM	-	4	GM	-	10	SM	-
			8	SM	-	10	SM	-	10	GM	-			

It can be seen that 7 of the 20 aircraft are scheduled to be repaired from which 5 have larger time since last repair values that the mean time between repairs and the other 2 have enough positive failure predictions that they are reasonable to be scheduled for repair. When looking at the aircraft which are not scheduled, one can see that 5 of the 12 aircraft are not scheduled even though their time since last repair is higher than the mean time between repairs. 2 of these (AC 8 and AC 14) do not have any available maintenance slots in the scheduling horizon. Furthermore, the time since last repair of AC 19 is only slightly above the mean time between repairs and therefore it is not striking if it is not scheduled for maintenance within the scheduling horizon of 10 days. On the contrary, AC 5 and AC 10 do have time since last repairs clearly higher than the average time between repairs, therefore, those need further analysis why they are not scheduled for maintenance. Table 2 shows that both aircraft have a specific maintenance slot on day 8 and that the aircraft scheduled for maintenance on day 8 (AC 13) has a time since last repair lower than AC 5 and AC 10. The analysis shows that in the final iteration, where the final maintenance schedule is determined the selection formula in equation (18) selects AC 13 because of the fewer amount of visits to the node even though the average awards of selecting AC 5 or AC 10 were higher.

In order to further evaluate whether the most appropriate aircraft are chosen to be maintained, the time since last repair information is analyzed. Table 3 splits the 20 aircraft into four t_r categories related to $t_{\bar{a}}$. The same is done for the scheduled aircraft. It can be seen that even though there is a large amount of aircraft with comparatively low t_r only few of them are scheduled, whereas clearly more of the aircraft with high t_r are actually scheduled for repair.

Table 3: Amount of AC in Optimization and amount of AC scheduled

	$t_r < 0.5t_a$	$0.5t_a < t_r < t_a$	$t_a < t_r < 1.5t_a$	$1.5t_a < t_r < 2t_a$	**Total**
Amount of AC in Optimization	4	5	4	7	20
Amount of AC Scheduled	1	1	2	3	7
Percentage of AC Scheduled	25 %	20 %	50 %	35 %	40 %

Analysis of the robustness of the maintenance scheduling results

In this section, we analyze the results of the maintenance scheduling model presented before. An important factor in the evaluation of the model is the robustness of the model. Meaning that during multiple runs with identical situations of aircraft characteristics (same t_r, $m_{i,j}$, $P_{i,j}$) the model should schedule the same or similar aircraft for repair. Table 4 shows the result of 20 runs with identical aircraft characteristics input. It can be seen that during half of the runs the same aircraft are consistently scheduled. Only on 5 of the runs, an additional aircraft (AC 19) is scheduled which is not done during the other 5 runs. The main reason for this is a time since last repair slightly above the mean time between repairs. The difference between the punishments of scheduling or not scheduling for maintenance are therefore close to each other which makes it difficult for the maintenance scheduling model to be consistent during random rollout simulations. There is furthermore a small difference for 2 of the aircraft (AC 4 and AC 9) on which days maintenance is scheduled, they do however switch their days, thus in total, the same days are chosen for repair actions.

Table 4: Heatmatrix Scheduled Repair Actions 20 Aircraft, 10 run

	AC 1	**AC 2**	**AC 3**	**AC 4**	**AC 5**	**AC 6**	**AC 7**	**AC 8**	**AC 9**	**AC 10**
Day 1	0	0	0	**4**	0	0	0	0	**6**	0
Day 2	0	0	0	0	0	0	**10**	0	0	0
Day 3	0	0	0	0	0	0	0	0	0	0
Day 4	0	0	0	**6**	0	0	0	0	**4**	0
Day 5	0	0	0	0	0	0	0	0	0	0
Day 6	0	0	0	0	0	0	0	0	0	0
Day 7	**10**	0	0	0	0	0	0	0	0	0
Day 8	0	0	0	0	0	0	0	0	0	0
Day 9	0	0	0	0	0	0	0	0	0	0
Day 10	0	0	0	0	0	0	0	0	0	0
	AC 11	**AC 12**	**AC 13**	**AC 14**	**AC 15**	**AC 16**	**AC 17**	**AC 18**	**AC 19**	**AC 20**
Day 1	0	0	0	0	0	0	0	0	0	0
Day 2	0	0	0	0	0	0	0	0	0	0
Day 3	0	0	0	0	0	0	**10**	0	0	0
Day 4	0	0	0	0	0	0	0	0	0	0
Day 5	0	0	0	0	0	**10**	0	0	0	0
Day 6	0	0	0	0	0	0	0	0	0	0
Day 7	0	0	0	0	0	0	0	0	0	0
Day 8	0	0	**10**	0	0	0	0	0	0	0
Day 9	0	0	0	0	0	0	0	0	0	0
Day 10	0	0	0	0	0	0	0	0	**5**	0

As can be seen in Figure 6 those promising robustness values can be found also for smaller sets of aircraft in the scheduling algorithm. In Figure 6 we define robustness as presented in equations (28) and (29), with an example calculation for the robustness of scheduling 20 aircraft. A robustness (choosing AC) of 100 % means that during all runs the same aircraft are scheduled.

And a robustness (choosing days) of for example 90 % means that on average 90 % of the scheduled aircraft are always scheduled on the same days during all runs.

$$\text{Robustness (choosing Days)} = \frac{\#\text{ AC remain scheduled on the same day for all runs}}{\#\text{ AC scheduled for maintenance during all runs}} \tag{28}$$

$$\text{Robustness (choosing AC)} = \frac{\#\text{ AC remain scheduled for all runs}}{\#\text{ AC scheduled for maintenance during all runs}} \tag{29}$$

As an example for the 20 aircraft presented in Table 4 and Figure 4, Robustness (choosing days) $= \frac{10+6+10+6+10+10+10+5}{75} = 89\%$ and Robustness (choosing AC) $= \frac{10+10+10+10+10+10+10+5}{75} = 93\%$.

It is striking that the robustness increases again for increasing fleet sizes. Comparing this to Table 5 shows that the same aircraft show less robust results when choosing days (AC 4 and AC 9) and therefore the increasing robustness can be explained by the fact that in a larger fleet size more aircraft are scheduled robustly. This means that robustness is not just linked to fleet size and the model but to individual aircraft characteristics.

Figure 4: Robustness of the model using different fleet sizes

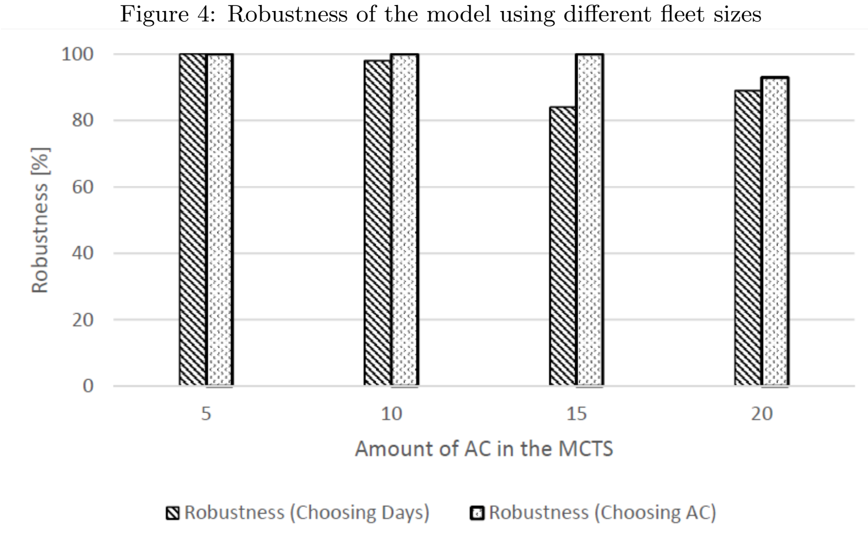


Table 5: Scheduled Repair Actions for different fleet sizes with 10 runs

	AC 1				AC 2				AC 3				AC 4				AC 5			
Fleet size	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**
Day 1	0	0	0	0	0	0	0	0	0	0	0	0	**10**	**9**	**4**	**4**	0	0	0	0
Day 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 4	**10**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**6**	**6**	0	0	0	0
Day 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 7	0	**10**	**10**	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Day 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

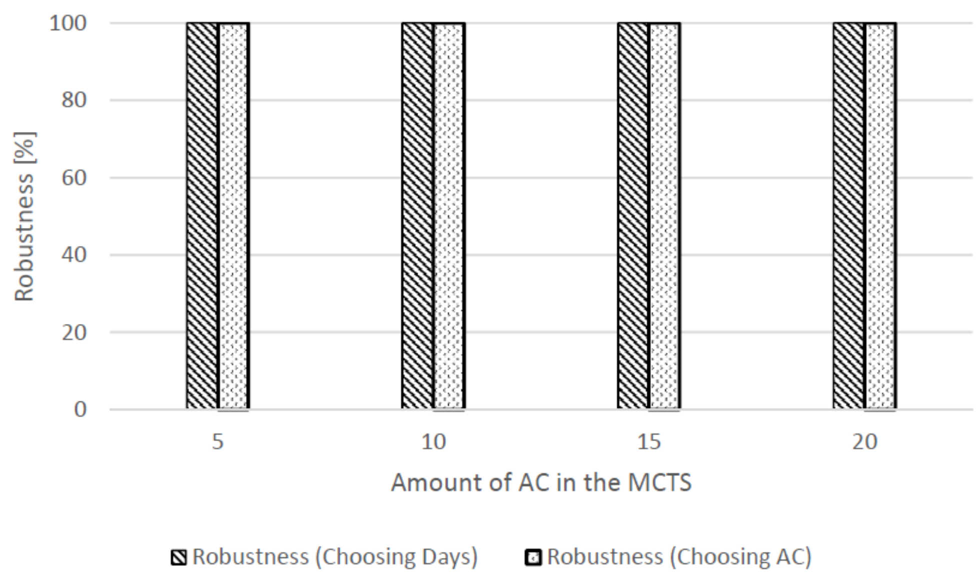
	AC 6				AC 7				AC 8				AC 9				AC 10			
Fleet size	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**
Day 1		0	0	0		0	0	0		0	0	0		**1**	**6**	**6**		0	0	0
Day 2		0	0	0		**10**	**10**	**10**		0	0	0		0	0	0		0	0	0
Day 3		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 4		0	0	0		0	0	0		0	0	0		**9**	**4**	**4**		0	0	0
Day 5		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 6		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 7		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 8		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 9		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
Day 10		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0

	AC 11				AC 12				AC 13				AC 14				AC 15			
Fleet size	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**
Day 1			0	0			0	0			0	0			0	0			0	0
Day 2			0	0			0	0			0	0			0	0			0	0
Day 3			0	0			0	0			0	0			0	0			0	0
Day 4			0	0			0	0			0	0			0	0			0	0
Day 5			0	0			0	0			0	0			0	0			0	0
Day 6			0	0			0	0			0	0			0	0			0	0
Day 7			0	0			0	0			0	0			0	0			0	0
Day 8			0	0			0	0			**10**	**10**			0	0			0	0
Day 9			0	0			0	0			0	0			0	0			0	0
Day 10			0	0			0	0			0	0			0	0			0	0

	AC 16				AC 17				AC 18				AC 19				AC 20			
Fleet size	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**	**5**	**10**	**15**	**20**
Day 1				0				0				0				0				0
Day 2				0				0				0				0				0
Day 3				0				**10**				0				0				0
Day 4				0				0				0				0				0
Day 5				**10**				0				0				0				0
Day 6				0				0				0				0				0
Day 7				0				0				0				0				0
Day 8				0				0				0				0				0
Day 9				0				0				0				0				0
Day 10				0				0				0				**5**				0

Instead of simulating random prognostics during the rollout it is also interesting to analyze the robustness of the maintenance scheduling model in case it receives a fixed set of prognostics also for future days. Which would be the situation if one would have more accurate prognostics such that a prediction about the prognostics can be made. As can be seen in Figure 5 the robustness is even better if one applies the same set of prognostics during all simulation iterations. During all runs identical aircraft are scheduled for maintenance on identical days of the scheduling horizon.

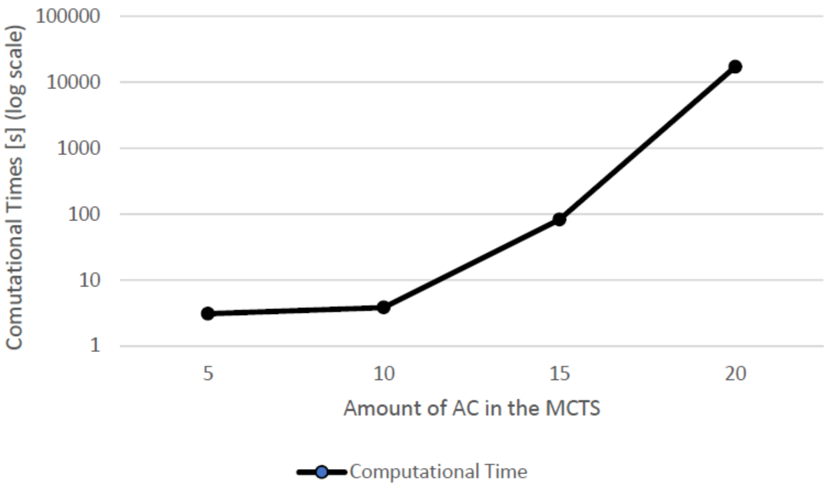
Figure 5: Robustness of the model using different fleet sizes, with fixed prognostics during simulation



Analysis of computational time of the maintenance scheduling model

Airlines operate many aircraft in their fleet and therefore it is interesting how well the model is able to schedule different amount of aircraft. As can be seen in Figure 6, the computational time highly increases with an increase of aircraft in the Monte-Carlo tree search. Up until approximately 12 aircraft the model is able to schedule the aircraft within 10 seconds or less. In order to schedule 20 aircraft, it requires however almost 5 hours, which is even exceeded by a computational runtime of 3.5 days when scheduling 25 aircraft. These runtimes were observed on a dual core 3.1 GHz Intel Xeon Platinum 8175 with 16 GB memory capacity and on a MATLAB 2018b version.

Figure 6: Computational Times of Different Amount of AC in the MCTS [s]



4.3. Sensitivity analysis of variable aircraft characteristics input

Table 3 previously showed the amount of aircraft scheduled categorized into time since last repair categories. The shown statistics are based on one run of the maintenance scheduling algorithm for the 20 AC aircraft characteristics input. Table 4 furthermore showed that multiple runs of that Monte-Carlo tree search algorithm using the same aircraft characteristics input schedule identical aircraft for maintenance. It is however also interesting to see whether that general trend is observable with different kind of aircraft characteristics input. Table 6 therefore shows the results of categorizations for multiple runs of scheduling 20 aircraft, each with a different set of input data. It can be seen that with random aircraft characteristics input the trend of scheduling more aircraft of the categories with $t_r > t_{\bar{a}}$ is even more visible than before.

Table 6: Amount of AC in Optimization and amount of AC scheduled (200 AC during 10 runs with random input)

	$t_r < 0.5t_a$	$0.5t_a < t_r < t_a$	$t_a < t_r < 1.5t_a$	$1.5t_a < t_r < 2t_a$	**Total**
Amount of AC in Optimization	45	50	54	51	200
Amount of AC Scheduled	12	21	35	38	106
Percentage of AC Scheduled	27 %	42 %	65 %	75 %	53 %

5. Additional Model Case Studies

The main model presented good and robust results when scheduling aircraft using classification prognostics and available maintenance slots. It is furthermore also interesting to see how the model behaves if probability prognostics are used instead of classification prognostics and how the model behaves if simulated prognostics of future days are not taken into account. This section presents both case studies by firstly presenting the changes to our maintenance scheduling model and secondly elaborating on the results and added benefits.

5.1. Failure distribution input

Changes to the model - Failure distribution input

Classification prognostics, as applied as an input to the main model of the Monte-Carlo tree search maintenance scheduling algorithm, are not the only available types of prognostics in the maintenance sector. This case study, therefore, focuses on implementing a failure distribution instead of classification prognostics. Meaning that at a given day a probability of failure is approximated. This probability can then be used in the punishment structure presented above instead of the accuracy values of NPV and PPV.

Failure probabilities can be calculated based on the failure function as can be seen in equation (30).

$$P_{i,j} = F(x) = 1 - exp(-\frac{1}{354} \cdot t_r) \tag{30}$$

The new version of the simulated predictions punishment uses the same concept as before. It punishes a simulated action if it does not match the simulated prediction made for that day. A probability ≥ 0.5 is considered a prediction to fail and a probability < 0.5 is considered as a prediction that the component will not fail. The new version of the second punishment value $C_{3',j,i,l}$ is defined on each layer j, $1 \leq j \leq L_d$ and for each aircraft i, $i \in I$.

$$C_{3',j,i,l} = \begin{cases} (1 - P_{i,j}) & \text{if } P_{i,j} < 0.5 \text{ and } a_{i,j,k_j} = 1 \\ 0 & \text{if } P_{i,j} < 0.5 \text{ and } a_{i,j,k_j} = 0 \\ P_{i,j} & \text{if } P_{i,j} \geq 0.5 \text{ and } a_{i,j_k;j} = 0 \\ 0 & \text{if } P_{i,j} \geq 0.5 \text{ and } a_{i,j,k_j} = 1 \end{cases} \tag{31}$$

The new version of the previous predictions punishment is able to include not just classification prognostics but also probability prognostics such that the punishment is stronger if the failure prediction is higher. The concept, however, is similar to the previous setup. If a failure prediction with a specific probability is made for the next 10 days and no repair is scheduled within those 10 days, a punishment equal to the probability is added at day 10. The new version of punishment $C_{4',j,i,l}$ is defined on each layer j, $1 \leq j \leq L_d$ depending on the actions a_{i,j,k_j} of the previous 10 days $j - 10, ..., j$, and for each aircraft i, $i \in I$.

$$C_{4',j,i,l} = \begin{cases} P_{i,j-10} & \text{if } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } P_{i,j-10} \geq 0.5 \\ 1 - P_{i,j-10} & \text{if } (S_{i,j,k_j} = 1 \text{ or } a_{i,j,k_j} = 1) \text{ and } P_{i,j-10} < 0.5 \\ 0 & \text{if } (S_{i,j,k_j} = 1 \text{ or } a_{i,j,k_j} = 1) \text{ and } P_{i,j-10} \geq 0.5 \\ 0 & \text{if } S_{i,j,k_j} = 0 \text{ and } a_{i,j,k_j} = 0 \text{ and } P_{i,j-10} < 0.5 \end{cases} \tag{32}$$

Numerical Results: Maintenance scheduling using component failure distribution

Failure probabilities add additional information to the model compared to classification prognostics. Which can also be seen in the results of the model extension with failure probability distribution as prognostics input. The same time since last repair and available maintenance slots input are used as before. Using equation (30) results in the following failure probability input for the days prior to initialization. As before in the main maintenance scheduling model, the failure probabilities for the future days are computed randomly in the simulations.

Table 7: Probability Prognostics Input

AC \ Day	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
1	0,378	0,380	0,381	0,383	0,385	0,387	0,388	0,390	0,392	0,393
2	0,485	0,487	0,488	0,489	0,491	0,492	0,494	0,495	0,497	0,498
3	0,467	0,469	0,470	0,472	0,473	0,475	0,476	0,478	0,479	0,481
4	0,839	0,840	0,840	0,841	0,841	0,841	0,842	0,842	0,843	0,843
5	0,774	0,774	0,775	0,776	0,776	0,777	0,777	0,778	0,779	0,779
6	0,424	0,425	0,427	0,428	0,430	0,432	0,433	0,435	0,436	0,438
7	0,695	0,696	0,696	0,697	0,698	0,699	0,700	0,701	0,702	0,702
8	0,778	0,779	0,779	0,780	0,781	0,781	0,782	0,782	0,783	0,784
9	0,811	0,812	0,812	0,813	0,813	0,814	0,814	0,815	0,815	0,816
10	0,831	0,831	0,832	0,832	0,833	0,833	0,834	0,834	0,835	0,835
11	0,688	0,689	0,689	0,690	0,691	0,692	0,693	0,694	0,695	0,696
12	0,104	0,107	0,109	0,112	0,114	0,117	0,119	0,122	0,124	0,127
13	0,336	0,338	0,340	0,342	0,344	0,345	0,347	0,349	0,351	0,353
14	0,819	0,820	0,820	0,821	0,821	0,822	0,823	0,823	0,824	0,824
15	0,552	0,553	0,554	0,555	0,557	0,558	0,559	0,560	0,562	0,563
16	0,727	0,727	0,728	0,729	0,730	0,730	0,731	0,732	0,733	0,733
17	0,786	0,786	0,787	0,787	0,788	0,789	0,789	0,790	0,790	0,791
18	0,374	0,376	0,378	0,380	0,381	0,383	0,385	0,387	0,388	0,390
19	0,661	0,662	0,663	0,664	0,665	0,666	0,667	0,668	0,669	0,670
20	0,351	0,353	0,355	0,356	0,358	0,360	0,362	0,364	0,365	0,367

Table 8 shows that 7 of the 8 aircraft scheduled in the original model are still scheduled for a repair action when applying a failure probability distribution as input instead of classification prognostics. Those aircraft are furthermore also scheduled on the same days as before. Special focus can, however, be put to the fact that again AC 19, with $t_r > t_{\bar{a}}$, is scheduled. This is again the same aircraft which already presented striking results during the robustness and sensitivity analysis in Table 5 and Table 10. This nicely shows that the main model with random rollout simulations is slightly unsure about scheduling AC 19 for maintenance but changes to the model (e.g. neglecting random prognostics simulations or adding probability distributions as prognostics) increase the certainty of the maintenance scheduling algorithm.

Table 8: Scheduled Repair Actions 20 Aircraft (main model and probability prognostics model extension)

AC	t_r	m_ij Day	m_ij Type	Repair? Main	Repair? Ex Prob
AC 1	t_r = 178	1	SM	-	-
		4	SM	-	-
		7	SM	✔	✔
		8	GM	-	-
AC 2	t_r = 245	8	SM	-	-
AC 3	t_r = 233				
AC 4	t_r = 657	1	SM	-	-
		4	SM	✔	✔
		5	GM	-	-
		7	SM	-	-
		8	SM	-	-
AC 5	t_r = 536	8	SM	-	-
AC 6	t_r = 205				
AC 7	t_r = 430	2	GM	✔	✔
		8	SM	-	-
AC 8	t_r = 543				
AC 9	t_r = 600	1	SM	✔	✔
		4	GM	-	-
		7	SM	-	-
AC 10	t_r = 639	8	SM	-	-
AC 11	t_r = 422	1	SM	-	-
		4	SM	-	-
		7	SM	-	-
AC 12	t_r = 49	1	GM	-	-
		4	SM	-	-
		7	SM	-	-
AC 13	t_r = 155	8	SM	✔	✔
		9	GM	-	-
AC 14	t_r = 616				
AC 15	t_r = 294	8	SM	-	-
AC 16	t_r = 469	5	SM	✔	✔
		8	SM	-	-
AC 17	t_r = 555	3	SM	✔	✔
		5	SM	-	-
		8	SM	-	-
AC 18	t_r = 176	1	SM	-	-
		3	GM	-	-
		10	SM	-	-
AC 19	t_r = 393	1	SM	-	-
		4	GM	-	-
		10	GM	-	✔
AC 20	t_r = 163	1	SM	-	-
		10	SM	-	-

5.2. 1 day future prognostics

Changes to the model - 1 day prognostics

As explained before we considered randomly simulated classification component prognostics during the simulation stage of the Monte-Carlo tree search due to the fact that the prognostics are not known to us and even with randomly selected prognostics input the main model is able to robustly schedule aircraft, only their robustness related to on which days the maintenance is scheduled is lower for increasing sets of aircraft. It is interesting to analyze the effect of not considering simulated prognostics. The same 20 aircraft as discussed before are scheduled for maintenance neglecting simulated prognostics for future days. Which means that the simulated prognostics punishment $C_{3,j,i,l}$ is not taken into account. The component classifications prognostics of day 1 which would be known at the moment of decision making are however taken into account and set fixed as presented in Table 9, which can be seen as an extension to Table 1.

Table 9: Classification Prognostics Input of Day 1

AC \ Day	1
1	1
2	0
3	0
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	0
13	0
14	1
15	0
16	1
17	0
18	1
19	1
20	0

Numerical Results: Without simulated prognostics of future days

Table 10 shows that the results are highly similar to the results of the main model presented in Table 2. The only difference is that the main model scheduled AC 19 only during some of the runs, whereas it is scheduled for maintenance when not considering simulated predictions. This shows that in the current setup of the maintenance scheduling model the simulated prognostics do not have a large effect on the results.

Table 10: Scheduled Repair Actions 20 Aircraft (main model and model without simulated prognostics)

AC	t_r	m_ij Day	m_ij Type	Repair? Main	Repair? no simP_ij
AC 1	t_r = 178	1	SM	-	-
		4	SM	-	-
		7	SM	✓	✓
		8	GM	-	-
AC 2	t_r = 245	8	SM	-	-
AC 3	t_r = 233				
AC 4	t_r = 657	1	SM	-	-
		4	SM	✓	✓
		5	GM	-	-
		7	SM	-	-
		8	SM	-	-
AC 5	t_r = 536	8	SM	-	-
AC 6	t_r = 205				
AC 7	t_r = 430	2	GM	✓	✓
		8	SM	-	-
AC 8	t_r = 543				
AC 9	t_r = 600	1	SM	✓	✓
		4	GM	-	-
		7	SM	-	-
AC 10	t_r = 639	8	SM	-	-
AC 11	t_r = 422	1	SM	-	-
		4	SM	-	-
		7	SM	-	-
AC 12	t_r = 49	1	GM	-	-
		4	SM	-	-
		7	SM	-	-
AC 13	t_r = 155	8	SM	✓	✓
		9	GM	-	-
AC 14	t_r = 616				
AC 15	t_r = 294	8	SM	-	-
AC 16	t_r = 469	5	SM	✓	✓
		8	SM	-	-
AC 17	t_r = 555	3	SM	✓	✓
		5	SM	-	-
		8	SM	-	-
AC 18	t_r = 176	1	SM	-	-
		3	GM	-	-
		10	SM	-	-
AC 19	t_r = 393	1	SM	-	-
		4	GM	-	-
		10	GM	-	✓
AC 20	t_r = 163	1	SM	-	-
		10	SM	-	-

The runtime analysis of the changed model resulted in almost identical results as before shown in Figure 6 for the main model.

6. Discussion

The Monte-Carlo tree search algorithm as an application in airline maintenance scheduling showed interesting results as presented above. In this section, we first discuss the results of the main maintenance scheduling model after which we discuss the results of the model extension using probability prognostics instead of classification prognostics.

6.1. Main maintenance scheduling model

The main objective of the model is to schedule multiple aircraft simultaneously for a pre-defined amount of days into the future. This needs to be done using classification prognostics and available maintenance slots into account. As shown in Table 2 this is indeed the results of the proposed model and as can be seen in Table 3 it is also done is a comparatively accurate manner. Most of the aircraft scheduled do have a time since last repair above the mean time between repairs. There are however also some aircraft which have a shorter time since last repair but are scheduled either way and others with a high time since last repair which are not scheduled.

Especially the impact of the prognostics is difficult to assess since the algorithm bases its choice on comparing many different branches and combinations in order to make a selection. This, unfortunately, makes it difficult to directly understand the choice of the model. It is, however, possible to find reasons for all aircraft why they are scheduled or why not.

A positive result of the main model is also the high robustness of the main maintenance scheduling model. Presenting the model with the same input always leads to a similar result. Meaning that the almost identical aircraft are scheduled and only sometimes deviates on which days this aircraft is scheduled. This constant good performance allows the assumption that the model is also able to robustly schedule larger amount of aircraft.

This is however hindered by the steeply increasing computational runtimes for larger amount of aircraft. The reason for this is the large increase of possible child nodes and therefore the possible options to evaluate. Up until around 20 AC of the same type the runtime is, however, feasible for an operational context. When looking at the fleet of different airlines one can see that many airlines operate less than 20 aircraft of the same type which means that this model would be feasible for a number of those cases.

Table 11: Fleet sizes of different European Airlines

Airline \ AC Type	A319	A320	A321	A330	A340	A350	A380	B737	B747	B777	B787
KLM				13				51	13	29	13
Lufthansa	27	94	65	16	32	14	14		32		
TAP	21	25	12	23	4						

The results of the main model also showed that the current implementations largely focus not just on the prognostics input but also highly on the time since last repair of the aircraft. This choice was deliberately made due to the low accuracy and availability of actual prognostics data. Table 10 showed that the scheduling results are almost identical without simulating predictions for upcoming days.

6.2. Additional Case Studies

As presented in Section 4 also the results of using failure distribution probabilities instead of classification prognostics in the maintenance scheduling model show promising potential. According to the time since last repairs of the scheduled aircraft the new approach is able to schedule 20 aircraft even more appropriately than before.

The practicality of the model is, however, an issue. The failure distribution probabilities are created based on an assumed failure function since they were not available for the component analyzed in the main model. We, however, assumed them to be correct without taking additional accuracy of those failure predictions into account.

The analysis neglecting simulated prognostics and only including previous prognostics and the ones from the first day showed that this version presents almost identical results and therefore making it as useful as the main model.

7. Conclusions and Recommendations

As stated before, our Monte-Carlo tree search maintenance scheduling algorithm is able to robustly schedule multiple aircraft based on their failure prognostics. The main issue of the current model and its' extension is, however, the run time. As stated in the literature review, many other scheduling models only start the planning process once an aircraft is triggered due to high time since last repair or low remaining useful life. The deliberate choice of considering all aircraft and letting the model compute which ones to repair ensures that all possible options are considered but is also the main reason for the high computational run times. The main recommendation is therefore to work on reducing the computational time for larger sets of aircraft. One might look into implementing some form of a trigger to remove aircraft not fulfilling any of the criteria required in order to be scheduled or otherwise increase searching efficiency. But it is also interesting to evaluate different selection procedures in the MCTS steps or to improve the current UCB1 procedure. The current implementation uses the default value of $\sqrt{2}$ as a factor between exploitation and exploration but other tuned factors might improve search efficiency while delivering similar maintenance schedules.

The extension offers an interesting solution for further developments. Probability prognostics create a more complete set of information about the health status of a component which makes it more useful for further implementations. It is however required to further improve the model in order to include the accuracy of the failure predictions which is not implemented in the maintenance scheduling model.

The case study when neglecting simulated prognostics showed good and almost identical results as the main model. It is therefore possible to use this change as the basis for future model improvements.

One possibility provided by the chosen Monte-Carlo tree search approach is the concept of a moving horizon. This is currently not yet implemented but would be interesting to analyze since it results in a more realistic solution. The current model once schedules all 20 aircraft for the upcoming ten days but a moving horizon scheduling approach would be able to make use of additional knowledge of days once they have passed. By scheduling each day for the upcoming ten days while implementing the new data of the last day would improve the usability and accuracy of the model further. Applying a moving horizon decision making process is also expected to solve some concerns of the robustness and not scheduling aircraft with high time since last repairs, since the algorithm only needs to be certain for the first day of the scheduling horizon.

Another aspect which should be analyzed if one wishes to improve real-life airline operations applicability is the fact that the current maintenance scheduling model assumes each maintenance slot be a full day and that is is always available to schedule an additional maintenance action, which is clearly a strong assumption.

A Monte-Carlo tree search approach was chosen due to the simplified prognostics input available and it proved to a suitable method for a simplified input environment. For future developments, one should aim however at obtaining more sophisticated prognostics data as an input to the planning framework. Linking actual and more accurate prognostics information and the maintenance availability of the respective aircraft one would be able to develop a more accurate model. This would then also make the implementation at an airline easier.

References

[1] P. Sparaco, LCCs on route to dominate european point-to-point travel, Aviation Week and Space Technology (New York) 173 (19) (2018) 46.

[2] X. Chen, S. Wang, B. Qiao, Q. Chen, Basic research on machinery fault diagnostics: Past, present, and future trends, Frontiers of Mechanical Engineering 13 (2) (2018) 264291.

[3] H. R. P. I. Report, Approaching zero downtime: The center for intelligent maintenance systems (IMS)n, Harbor Research Inc. (2003) 1–11.

[4] T. Grubic, L. Redding, T. Baines, D. Julien, The adoption and use of diagnostic and prognostic technology within UK-based manufacturers, The Journal of Engineering Manufacture 225 (8) (2011) 1457–1470.

[5] X. Lei, P. Sandborn, Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements, Renewable Energy 118 (Part B) (2018) 188–198.

[6] M. You, G. Meng, A predictive maintenance scheduling framework utilizing residual life prediction information, Journal of Process Mechanical Engineering 227 (3) (2012) 185–197.

[7] B. Zhang, L. Xu, Y. Chen, A. Li, Remaining useful life based maintenance policy for deteriorating systems subject to continuous degradation and shock, in: 51st CIRP Conference on Manufacturing Systems, 2018.

[8] F. Camci, Maintenance scheduling of geographically distributed assets with prognostics information, European Journal of Operational Research 245 (2) (2015) 506–516.

[9] F. Camci, System maintenance scheduling with prognostics information using generic algorithm, IEEE Transactions on Reliability 58 (3) (2009) 539–552.

[10] Z. Li, J. Guo, R. Zhou, Maintenance scheduling optimization based on reliability and prognostics information, in: Annual Reliability and Maintainability Symposium (RAMS), 2016.

[11] S. Zhang, M. Du, J. Tong, Y.-F. Li, Multi-objective optimization of maintenance program in multi-unit nuclear power plant sites, Reliability Engineering and System Safety 188 (1) (2019) 532–548.

[12] S. Lakshminarayanan, D. Kaur, Optimal maintenance scheduling of generator units using discrete integer cuckoo search optimization algorithm, Swarm and Evolutionary Computation 42 (1) (2018) 89–98.

[13] H. Yamashina, S. Otani, Optimal preventive maintenance planning for multiple elevators, Journal of Quality in Maintenance Engineering 7 (2) (2001) 128–150.

[14] G. Chaslot, J.Saito, B.Bouzy, J.Uiterwijk, H. V. D. Herik, Monte-Carlo strategies for computer go, in: BeNeLux Conference on Artificial Intelligence, 2006, pp. 83–91.

[15] L. Kocsis, C. Szepesvari, Bandit based Monte-Carlo planning, in: European Conference of Machine Learning, 2006, pp. 282–293.

[16] M. FU, Alphago and Monte Carlo tree search: The simulation optimization perspective, in: Winter Simulation Conference, 2016, pp. 659–669.

[17] S. Gelly, A contribution to reinforcement learning; Application to computer-Go, Univeristy Paris-Sud, Paris, France, 2007.

[18] G. Chaslot, C. Fiter, J. Hoock, A. Rimmel, O. Teytoud, Adding expert knowledge and exploration in Monte-Carlo tree search, in: Advanced Computing Games, 2010, pp. 1–13.

[19] Y. Osaki, K. Shibahara, Y. Tajima, Y. Kotani, An Othello evaluation function based on temporal difference learning using probability of winning, in: Symposium on Computational Intelligence and Games, 2008, pp. 205–211.

[20] P. Hingston, M. Masek, Experiments with Monte Carlo Othello, in: Congress on Evolutionary Computation, 2007, pp. 4059–4064.

[21] A. Rimmel, F. Teytoud, T. Cazenave, Optimization of the nested0 Monte-Carlo algorithm on the traveling salesman problem with time windows, in: European Conference on the Applications of Evolutionary Computation, 2011, pp. 501–510.

[22] Z. Bnaya, A. Felner, S. Shimony, D. Fried, O. Maksin, Repeated-task Canadian traveler problem, in: Symposium on Combinatorial Search, 2011, pp. 24–30.

[23] H. Liu, K. Austin, M. Forbes, M. Kearney, Monte-Carlo tree search in dragline operation planning, IEEE robotics and automation letters 3 (1) (2018) 419–425.

[24] N. Zhao, Y. Guo, T. Xiang, M. Xia, Y. Shen, C. Mi, Container ship stowage based on Monte Carlo tree search, Advances in Sustainable Port and Ocean Engineering. Journal of Coastal Research 83 (1) (2018) 540–547.

[25] H. Nakhost, M. Mller, Monte-Carlo exploration for deterministic planing, in: International Joint Conference of Artificial Intelligence, 2009, pp. 1766–1771.

[26] D. Silver, G. Tesauro, Monte-Carlo planning in large POMDPs, in: Annual Conference on Neural Information Processing Systems, 2010, pp. 1–9.

[27] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of Monte Carlo tree search methods, IEEE Transactions on computational intelligence and AI in games 4 (1) (2012) 1–43.

[28] D. Altman, J. Bland, Statistics notes: Diagnostic tests 2: Predictive values, BMJ 309 (6947) (1994) 102. doi:10.1136/bmj.309.6947.102.

[29] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Machine learning 47 (2) (2002) 235–256.

[30] T. Matteson, Airline experience with reliability-centered maintenance, Nuclear Engineering and Design 89 (2) (1985) 385–390.

II

Literature Study (previously graded under AE4020)

2 Prognostics

Prognostics and health management (PHM) is an engineering profession which aims at predicting the time of failure of a system or component based on monitoring the health status or degradation status of a component. Related to that is the prediction of the remaining useful life (RUL) or failure probability. There are a variety of approaches and categories associated with PHM applications. The following chapter will first elaborate on the process and the current state of the art with respect to prognostic methods in Section 2.1 after which the more specific topic of multi-components or multi-units in prognostics will be covered in Section 2.2. Finally, Section 2.3 presents the trends and developments with respect to PHM and evaluates possible improvements.

2.1. Prognostics Methods

The field of prognostics and health management is a growing specialty due to the fact that it offers the opportunity to gain more information about the health status of the equipment. Knowing how well the equipment is performing and possibly for how long it will be working and applying that knowledge correctly enables companies to save costs and to increase their operational up-time. This section gives an insight into the general concepts of prognostics and about how they are obtained.

2.1.1. On-line and Off-line PHM

Literature generally differentiates between two types of prognostics and health management concepts. Namely on-line PHM (real-time PHM) and off-line PHM. As the names already indicate the first uses live data to assess the health status of a system whereas the second works with backed-up sensor data.

On-line monitoring is mostly applied in cases of mission critical systems or applications of high value products. Common examples are on-board computers using real-time sensor data in cars or unmanned vehicles to provide range distances, to re-plan missions or to reconfigure control settings. Another focus point of real-time PHM is to test and verify whether all electronic systems are operating correctly. This is done either instead of human inspections or as a support to the user or technicians by simplifying the inspection task. The main downside of on-line prognostic and health management is the large amount of required computational capacity of the on-board computers.

Since most prognostics require extensive simulations and computations, a lot of complex prognostics computations are often done off-line. For this method a variety of sensor data is selected and collected from the system and off-line PHM computer simulations are then used to predict for example the RUL of that system. [3]

2.1.2. Modeling Approaches

One of the main decision aspects when working with PHM is to actually develop predictions about for example the remaining useful life of the component. Three general concepts of obtaining prognostics are currently known. The modeling approach either focuses on a data-driven methodology

[3], a experience-based model [4] or a model-driven approach [5]. It is however also possible to combine those three in a hybrid modeling approach. A variety of considerations need to be made in order to decide which of those strategies should be implemented in the PHM approach.

Experience-based Prognostics

This form of prognostics is not considered in all of the publications available. Some solely focus on model-based and data-driven prognostics [3]. Other like [4, 6] classify it as one of the three main methods.

The experience-based approach is comparable to a modeled representation of the actual logical steps and work done by a system specialist. Based on the experts opinion a number of *IF-THEN* rules are defined to draw conclusions about the current health state of the system [6]. An expert strategy using a knowledge data base and *IF-THEN* rules were determined and proposed for example by Biagetti and Sciubba [7] related to a gas turbine-based cogeneration system. It is however as explained below often more common to focus on smaller components due to the interdependencies in larger systems. Using an experts opinion offers the possibility of implementing the experience of multiple years or even generations. However the main drawback of this approach is the limitation with respect to the complexity of the system. Those expert rules are often not able to consider a larger amount of combinations or dependencies and rely heavily on the opinion of a small group of experts which might result in an incomplete set of rules and consideration of characteristics [4]. It is furthermore very complicated to define accurate numerical predictions of the remaining useful life. Experience-based prognostics are therefore more often used to obtain an indication about the health status of a specific component or equipment instead of retrieving an accurate number of remaining useful life. If an appropriate component of which a large amount of known interdependencies, behavioral characteristics and fault-consequence combinations is chosen the result of the prognostics algorithm can be comparatively accurate [8]. Majidian and Saidi [8] developed experience-based and neural network (form of data-based prognostics) RUL predictions of boiler re-heater tubes and compared them with each other. The results showed that both estimations were rather close.

Model-based Prognostics

To apply a model-based prognostics approach one requires the detailed understanding of the system at stake. That means that in some form a physical representation of that system needs to be created in order for it to be analysed. There are a variety of options to do so [9–13] each related and customized to its specific real life component application. It is for example possible to model fatigue crack dynamics using a nonlinear stochastic approach [14] or to use first and second-order nonlinear differential equations to represent damage accumulation in a structural dynamic system [15]. Chelidze [16] furthermore developed a prognostics approach based on a mathematical model representing an electro-mechanical system consisting of a cantilever beam oscillating due to the potential fields of two permanent magnets and electromagnets powered by batteries.

The health assessment is then based on the difference between measured data of an actual system compared to the output of the theoretical healthy model of the system about how it should behave. A large difference is therefore related to a health degradation of the system. In order to define when a malfunction should be detected, a threshold needs to be defined. [17]

One of the main advantages of the model-based approach is the close relation to the actual physical system. It means that small changes in the system can rather easily be incorporated into the model without actually changing the prognostics algorithm. Furthermore the model can be improved as soon as more insights about the degradation and failure processes are known. [17]

An important disadvantage of model-based prognostics is however that it is very complex and almost impossible to fully understand, model and simulate a whole multi-component system. Often simplifications need to be made or behaviour needs to be separated which creates some limitations to the completeness of the prognostics. Those simplifications create a somewhat imaginary system which makes it difficult to link and compare it to real-life applications in the industry.

Data-driven Prognostics

One option to create prognostics for complex systems without understanding the detailed physical set-up of the system is applying data-driven prognostics. This method was enabled by the technology advancements within the field of modern sensor systems and data storage and processing technologies [18]. The large amount of obtained data is then the basis for data-driven algorithms which analyze the sensor data to monitor the health status and to predict failure of the system [3].

One of the main disadvantages of the data-driven approach is the amount of data required to feed and test the algorithms. In an optimal scenario large number of run-to-failure data sets of the system in question are wanted. This is however a time consuming and expensive process [19] and therefore often an alternative needs to be found to train and validate the algorithms.

In the early development of prognostics the most used approach was the reliability analysis based on historical data process modeling. A large downside of those however were the static and rigid results due to the historic information [20, 21]. Therefore new methods were developed in order to cope with more dynamic and flexible systems and data sets. Generally speaking there are currently two commonly used data-driven approaches within prognostics and health management. Namely a statistical approach and a machine learning approach. [22]

Statistical Approach

As the name statistical approach already indicates, this method is set up by analyzing fundamental statistical parameters like mean, variance and median. Hereby a probability density function (PDF) of the functional operational data set is created and new incoming data points can be compared to that nominal PDF to detect anomalies. This straightforward comparison between new data points and nominal probability function offers the possibility to give realistic confidence intervals to use in the following decision making process. Those confidence intervals are the main advantage of the statistical approach if the assumed probability density function is indeed a valid representation of the actual system behaviour. If however the assumed statistical characteristics are not a good fit with respect to the actual system behaviour, the results of the algorithm will present anomalies which are not actual errors. Meaning that results require careful evaluation and interpretation before any possible usage. [3]

There are two methods of formulating the statistical characteristics of the system. Parametric approaches assume those statistical properties based on commonly known probability distributions and use the data set to calculate the distribution parameters. The non-parametric approach however does not take any existing probability distributions into account. It is therefore also suited in case the underlying distribution is not known or if the data does not fit to any known PDF.

Sutharssan et al. [3] defined a variety of possible statistical approaches. The extreme value theory (EVT) for example sets a threshold extreme value as an anomaly detection as applied in [23, 24]. In contrast to that is it also possible to determine the maximum-likelihood estimation to map input data accordingly [22].

Machine-learning Approach

During the past years a lot of work has been done with respect to machine learning which is a subfield of artificial intelligence. Those algorithms are trained using a training data set from which the algorithm learns the behavioral properties of the system. That knowledge is then further used to cluster future data into healthy or anomaly. The main advantage of machine learning with respect to the statistical approach is that the data relationships do not need to be pre-defined which allows it to be used in a variety of complex systems with potentially unknown physical context. [25]

The two main categories of machine learning in failure prognostics are supervised learning and unsupervised learning. Where the main difference can be identified as whether or not a labelled output for the training data set is provided [3]. In case of the supervised approach, the algorithm will be fed with a data set including output labels according to a classification scheme. Its goal is then to learn the relationship between input and output labels such that it can predict the output labels of new incoming input labels. Those output labels are not available in an unsupervised learning approach. The main application of this approach is therefore to detect patterns in an existing input data set and to classify them. In order to do so a set of healthy test data is given to the algorithm to understand the characteristics of healthy data after which it will be able to distinguish between healthy and anomaly data. [3]
It is furthermore possible to make a distinction between the predictive modeling outcomes within machine learning, namely a classification predictive modeling and a regression predictive modeling. Using a classification approach creates a mapping function which is able to label new incoming input. A fail/not fail prediction is often called a two-class or binary classification but it is also possible to distinguish between more than two classes. A regression approach however develops a function to predict a continuous output variable. In the prognostics context this could be related to a RUL prediction in days or cycles. [26]

The most commonly used machine learning approach in prognostics are neural networks but also Support Vector Machines (SVM), Gaussian processes and Bayesian networks [3].
Neural networks as applied by Schwabacher [25, 27] are trained to optimize the network parameters by minimizing the error compared between the input and prediction.
Support vector machines originate from pattern recognition and are used in prognostics by learning and defining of normal operating region with which new incoming data can be compared as done in [28, 29].
Gaussian process regression as applied in [30, 31] define multiple random variables to create a distribution representing the distribution of input values.
When using a Bayesian network approach as done in [32, 33] the conditional probabilities of possible nodes (related to the input) depending on other nodes is determined.

Combining Data-driven Algorithms
As stated in the paragraphs above all of those data-driven approaches have some downsides. In most cases multiple data-driven algorithms are developed to evaluate which one is most suited. This however has some crucial disadvantages, namely that the different algorithms are potentially not robust, a lot of time and effort is invested into algorithms which are not used after all and it requires not only a training set but also a testing set to validate which one is most suited. It is possible to develop a combination of a variety of data-driven prognostic approaches connected by a weighted-sum method [18]. It was found that the result of this combination resulted in more accurate remaining useful life estimations than any of the individual data-driven algorithms. This strategy of combining multiple data-driven models can also be considered as a form of hybrid approach as further elaborated on below.

Hybrid Approaches
Up until a few years ago the above mentioned approaches have mainly been used individually but recently a number of studies combined experience-based, model-based and data driven approaches to make use of the best combination of advantages. The following combinations can be defined [4]:

- Experience-based model + data-driven model (as applied in [34, 35])
- Experience-based model + model-based approach (as applied in [36, 37])

- Data-driven model + data-driven model (as applied in [38–40])
- Data-driven model + model-based approach (as applied in [41–43])
- Experience-based model + data-driven model + model-based approach (as applied in [44, 45])

The most common combination is the one between data-driven and model-based approaches. Where the model-based approach is used to apply general physical laws and to identify and update model parameters from the incoming test data. Whereas the data-driven model is used to predict anomalies which then again triggers the actual RUL prediction process.

Overall three challenges need to be overcome in order to implement the hybrid approach efficiently. Firstly the best combination of models needs to be selected, depending on the system in question, the available data, the required output and further considerations. Secondly once some models are selected a method to combine them is required. This can also be related to the situation at hand and which approach is chosen for which task. Thirdly a combination of models can result in unexpected uncertainties, sensitivities of other performance indicators. It is important to manage those prediction performances appropriately in order to draw useful conclusions from the model. [4]

2.2. Prognostics of Multi-Components

As stated above, the field of prognostics is growing rapidly but unfortunately the specific field of prognostics of complex multi-components is still an under explored area. The main challenge is to fully understand and implement the interdependencies between multiple components [46]. There is actually more research done about implementing multi-components in a prognostic maintenance schedule than looking at the individual prognostics of multi-components. Section 4.2 later onward presents more insight into the implementation of multi-components into maintenance planning optimization.

The reason that it is useful to implement multi-components into prognostics is the fact that most systems actually consists of multiple components instead of just one. The multiple components in those larger systems are usually interdependent on various levels. Literature generally classifies between three dependencies: stochastic dependence, structural dependence and economic dependence. [47]
Stochastic dependence can be described as components being related to each other by their degradation level, in that case a heavy deterioration of one component induces increased degradation of a dependent component. If components are physically connected they usually form a structural dependence. This means that the maintenance technician will be required to act on one component to reach the component which actually needs the maintenance action. The economic dependence can be further classified into positive economic dependence (PED) and negative economic dependence (NED). If replacing or maintaining multiple components at the same time is less expensive than the sum of their individual maintenance it can be defined as a positive economic dependence. Whereas it is NED if the cost increases if multiple maintenance tasks are performed simultaneously compared to the same separate activities. [48]
In order to develop prognostics or more specifically RUL estimations of one component in a larger system it is crucial to understand as many interdependencies as possible. Ribot et al. [49] developed a multi-component prognostics tool focusing on systems with only a few interdependencies between individual components making it less useful in more complex systems.

One possibility to handle this multi-component problem combines individual failure functions of multiple components and connecting their interdependencies in order to obtain an overall RUL estimation [46]. This study starts with defining failure probability density function for all individual components with a Weibull model representation of each probability function. Those functions are further developed into individual cumulative distribution functions and linked to component degradation states. Finally the interaction level is incorporated between the multiple components resulting in a change of RUL estimation of individual components.

Another study focused on analysing factors which influence multiple components in a complex system since they might be affected by the same environmental aspects [50]. The research uses multiple component sensor data of a complex system to estimate the degradation state by filtering out the noise of the environmental factors. Finally the RUL of individual components based on their interdependent factors is determined from the degradation function.

When looking at multi-component prognostics the main focus of most studies is to develop prognostics for a larger system consisting of multiple components. An important aspect to consider is however also the fact that an operator can have multiple units of a specific system. Having multiple units of a component is incorporated in a way that the amount of health and possible failure data is increased. This means that algorithms or models are often based not just on the data of one single- or multi-component but on multiple units of component. The knowledge gained over all units is then used to predict the RUL or other PHM information for a specific single- or multi-component.

2.3. Diagnostics and Prognostics State of the Art

Prognostics and health management approaches started mainly as a form of diagnostics, meaning that researchers focused initially at looking backwards. Diagnostics tools were developed in order to determine failure modes and different correlations between factors which might have caused a specific failure. A tremendous amount of progress has been made in recent years in order to diagnose faults and errors but research in the field is not yet complete. It is possible to define a number of challenges within the diagnostics field which need to be handled in the next years to cope with the changing and growing requirements of the industry. Those challenges are changing from behavioral research to mechanism studies, from qualitative to quantitative research, from single to group fault research, from severe to weak fault research and from component to system-level fault research [2]. This means that PHM research reached a point where evaluating a system in depth is possible. It also includes that the actual mechanism instead of the reaction of the system is analysed in order to draw conclusions about the overall performance.

After initial successes over the years more and more studies were performed to use the gained knowledge about the past to predict future failures as a form of prognostics. The start of research in this field was rather tedious and slow due to the lack of gathered complete data and computational capacities. Initially most prognostics methods were based on an experience-based approach. As explained above those algorithms were able to implement the years of experience of many experts and did not require an extensive health monitoring data set, but were however limited to simpler systems with few interdependencies.
Within time more sensors were implemented in on-board systems, and the increasing amount of data allowed the use of more sophisticated methods to predict the remaining useful life of a component. Changing from experience based models to physics based approaches and finally to data-driven models of the degradation process of components.
Using sensor data as an input for RUL predictions, however results in another challenge for PHM researchers. The data does not simply state the degradation state of a component but presents measurements of some other forms of indicators like temperature, vibrations, noise level or displacement. This data however includes uncertainties and noise. Therefore the data needs to be prepared using a variety of filters or stochastic methods. [51]

Even though a lot of research has been conducted in the field of PHM, the results of most prognostics algorithms are often still not exact enough to use them in the real-life decision process. This is due to the fact that the industry still struggles to collect the amount of data needed to optimally use the algorithms on a variety of components. Even using the advantage of multiple units is currently not sufficient to obtain enough health data for all systems. Up until now most approaches focus on individual components or smaller systems instead of all larger systems or an overall aircraft, simply because there is not yet an always valid and useful prognostics approach. A key aspect when developing prognostics is to choose the appropriate method and algorithm which results in the difficulty to determine a successful PHM application for an overall system [3].
Besides further improving the fault diagnosis as explained in the beginning of this section it is therefore crucial for researchers to improve the applicability of prognostics. It should be the goal to develop prognostics algorithms which are able to predict the remaining useful life of complex systems including as many interdependencies as possible to make it easily comparable to real life systems.

3
Airline Maintenance Schedules

Airline maintenance schedules consist of a variety of aspects which need to be considered in the planning of maintenance actions. The following chapter will give a general overview of previous and current maintenance strategies including a description of the variety of common types of maintenance in the aviation sector in Section 3.1. This is followed up by an elaboration of current research done in the field of maintenance scheduling in Section 3.2. An elaboration of the cost factors in airline maintenance is presented in Section 3.3. Finally Section 3.4 describes the current status of real-life airline maintenance schedules and the difference with respect to the scientific progress.

3.1. General Airline Maintenance Methods

Airline maintenance schedules consist of a large amount of tasks and procedures which need to be planned and performed. Maintenance scheduling is the process of defining maintenance opportunities meaning that an aircraft is scheduled for maintenance on a specific day or time. Maintenance planning is the next step where the actual tasks and activities are chosen to be done during a specific maintenance opportunity. This section presents the most important strategies, developments and aspects of airline maintenance operations.

3.1.1. History of Airline Maintenance

The maintenance strategies in the airline industry changed a lot from the beginning of manned flight up until today. In the beginning not a lot of focus was put into scheduling or planning of maintenance. This corrective maintenance strategy meant that no actions were performed unit a component had failed. Therefore each component was fully utilized but it however resulted in high cost and down-times. The unexpected breakdowns were furthermore a large safety hazard.

Within the 1930's until 1950's researchers analysed failures and how and why they occurred. The result of this research was the bathtub curve as can be seen in Figure 3.1. The response of the industry was to implement preventive maintenance in order to repair or replace components before they reach the wear-out failure phase and therefore before the actual failure occurred. Components were replaced based on hard-time or on-condition moments. Hard-time maintenance used early results of statistics and reliability analysis to estimate when components start to wear-out. Based on some general calculations fixed intervals were determined at which those kind or components require repair or replacement. The intervals were based either on calender days, flight cycles or flight hours. When applying on-condition maintenance the component in question is measured and compared to a set of standard thresholds. Whenever a measurement appears to be insufficient compared to the set standard, a repair or replacement action is executed or scheduled. Preventive maintenance highly increased the safety of flight by more often preventing in-flight failures but it was and still is difficult to predict when to optimally replace a component and therefore useful component life was not used, resulting in higher cost for the operator.

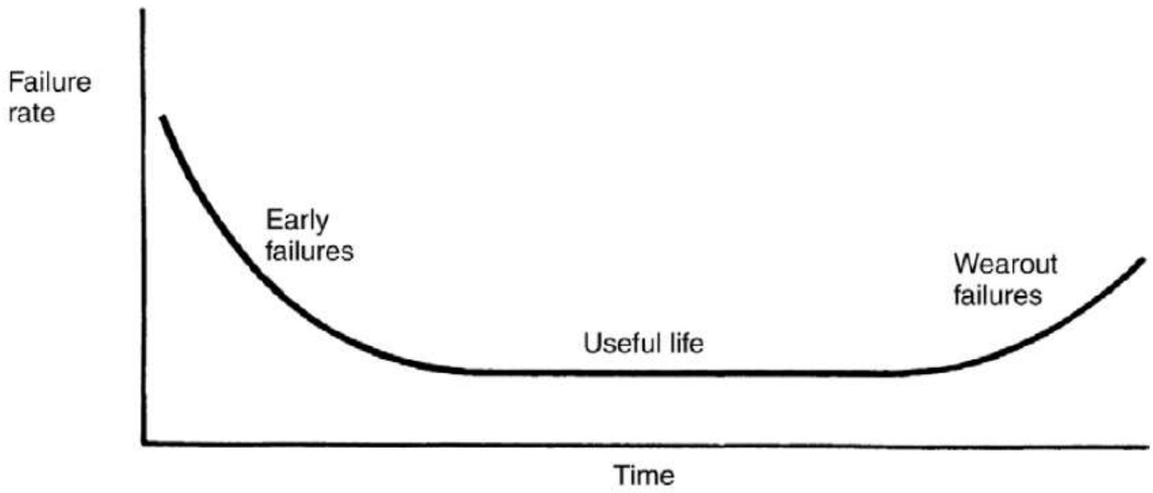


Figure 3.1: Schematic Drawing of the Failure Rate (Bathtub Curve)

In order to reduce the cost of not-utilized component life time more research had been conducted. Researchers found in the late 1960s that only six percent of the complex system actually show wear-out behaviour at the end of their life time [52] as can be seen in Figure 3.2. Researchers and in the industry responded to those finding by developing a new maintenance strategy called predictive maintenance. It includes the method of on-condition maintenance which was introduced in preventive maintenance but also introduces a new predictive aspect. By predicting when a component actually fails unexpected failures can be prevented and repairs and replacements can be scheduled as convenient and cost effective as possible. While at the same time minimized wasting useful lifetime of functional equipment by replacing those too early. It however requires sophisticated monitoring data collecting systems and prognostic and health management tools as explained before in Chapter 2.

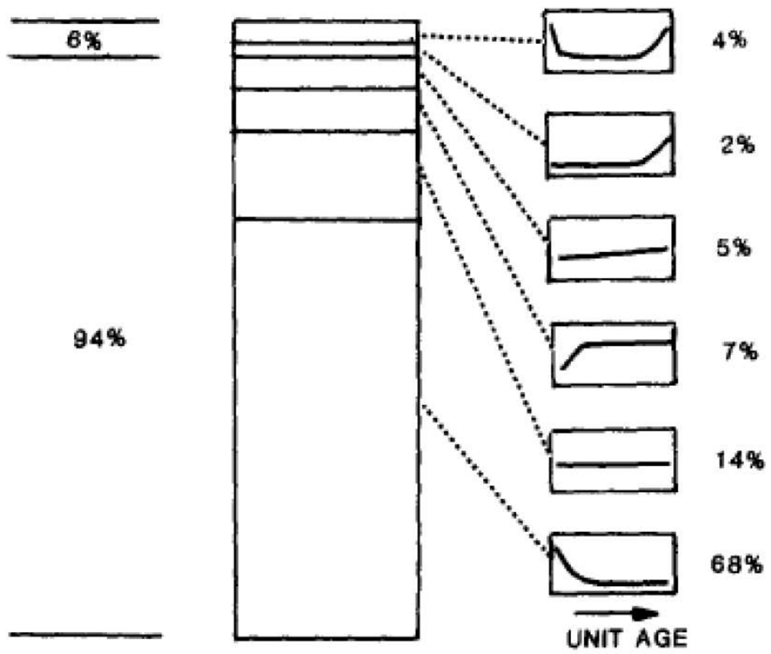


Figure 3.2: Experiences Age-Reliability Relationships [52]

3.1.2. Types of Airline Maintenance

Airline maintenance consists of a variety of maintenance actions, which can be divided into scheduled and unscheduled maintenance tasks. The Maintenance Steering Group (MSG) furthermore developed multiple maintenance decision support tools which changed related to the above mentioned change in maintenance strategies. This section will elaborate on those three aspects.

Scheduled (routine) maintenance

A lot of checks are scheduled by the maintenance operator based on international regulations and safety requirements. Those tasks are then usually planned either during line maintenance, base maintenance or shop maintenance. An overview of the most common types of airline maintenance can be found in Table 3.1.

Table 3.1: Overview of Common Types of Airline Maintenance

Type	**When and Where**	**Common Tasks**	**Duration**	**Remarks**
Transit Check	·Each turn-around · Line Maintenance	· Check for obvious damage · Service engine oil as required	TAT	
Daily Check	· Before first flight of the day or · when aircraft is on ground for more than 4 hours · Line Maintenance	· Operational checks of TCAS, emergency lights, stand-by power · Checking condition of landing gear, brakes, oil levels		
A-Check	· 400-600 flight hours or · 200-300 flight cycles · Base Maintenance	· Inspections of interior/exterior · operational checks · oil and filter checks and servicing	· 1 day · 50-70 man hours	B-check tasks are included
C-Check	· 12-24 months · Base Maintenance	· Functional and operational system checks · cleaning and servicing · Service Bulletins	1-2 weeks	Includes A-Check (and B-Check)
D-Check	· 6-12 years · Base Maintenance	· Removing exterior · Inspection, repair, replacement of internal structures and systems	2 months	Includes A-, B- and C-Check

Line Maintenance Checks

Most of the activities planned during line maintenance consist of recurrent (service) checks. The crew performs pre- and post-flight visual inspections in order to notify the maintenance operator

about changes, problems or anomalies. The check itself belongs to the scheduled activities, the repair however in most cases not since it does not fit in the allocated time frame of the respective check. Turnaround or transit checks include crucial scheduled task which are done during each turn-around time (TAT) as a form of line maintenance. Most of the activities associated are checks for obvious visual damage or basic servicing actions. The daily or 48 hour check already includes more work intensive actions but it can however still be performed during line maintenance. Scheduled activities can be categorized as operational checks, testing and physical checks of safety related components like landing gears, tires, breaks, emergency lights, TCAS (Traffic Alert and Collision Avoidance System).
Larger scheduled maintenance tasks are usually performed during letter checks as a form of hangar maintenance. Those letter checks are mostly planned after a fixed amount of flight hours or flight cycle interval depending on the regulations specific to the aircraft type.

A-Checks

A-checks are the smallest checks and occur approximately every 400-600 flight hours or 200-300 cycles. The planned actions consist of some general inspections or operational checks but also small lubrication, filter and fluid replacements. Activities planned during A-checks require the aircraft to be approximately one day on the ground with 50-70 man hours of maintenance actions. Lately maintenance operators developed different sizes of A-checks and therefore including most of the B-check tasks in the larger A-checks.

C-Checks

After 12-24 months each aircraft requires a larger C-check which is an extensive list of tasks. During a C-check also all A- and B-check actions are performed but furthermore different operational and functional not only of components but overall systems are executed. The interior and exterior are also cleaned and serviced. An important focus of scheduled C-check tasks are also Service Bulletins (SB) provided about the specific aircraft type. The usual duration of such a check is around 1-2 weeks.

D-Checks

Finally the largest letter check is the D-check which is scheduled only every 6-12 years and it includes a large dismantling effort in order to reach the internal structure of the aircraft. This allows the technicians to maintain, repair or replace the internal components which are otherwise not reachable from the outside. This check can take up to two months to complete. Logically due to the high work load a large amount of cost are related to a D-check. Therefore airlines wish to reduce the number of D-checks as much as possible by for example selling an aircraft before a new check is required.

Those checks are scheduled based on planned and forecasted flight hours and flight cycles in correlation with the MSG approaches as explained below. This results in a preliminary schedule which is basis for a continuous improvement based on the actual maintenance performance, inspection results and detailed flight schedules.
Currently a number of airlines already stopped applying a letter check scheduling frame but to use a more flexible maintenance scheduling. This is supported by the MSG-3 approach as explained below and further research in the field improvements of base maintenance as elaborated on in Section 3.2.

Unscheduled (non-routine) maintenance

The above mentioned maintenance actions are in almost all cases related to flight cycles or time intervals. All inspections and tasks required due to other unexpected situations are usually classified as unscheduled maintenance. Examples of those non-routine checks are heavy and unusual flight situations like heavy turbulence, lightning or bird strike or not optimal landing manoeuvres. If those inspections show that maintenance actions are required the technicians have a few options.
One option is to postpone the repair of the specific component if it is allowed according to the minimum equipment list (MEL). This list defines aircraft specific requirements about which components need to be fully functional and without which a flight can still take place. Usually some conditions are mandatory to be fulfilled in order to defer the repair of a maintenance action.

In case a system with redundancy fails it is often possible to postpone the maintenance. Regulations state that this is only possible if the operator is aware of the breakdown of one of the systems and if serviceability of the system is performed.
Lastly maintenance operators developed a fast way of performing maintenance for some of the aircraft systems. Line replacement units (LRU) might be used to easily replace a system with a working unit after which the faulty system is repaired in the workshop. This allows for fast maintenance actions especially during turn-around time or line maintenance without delaying the aircraft any further.

The development of generally more flexible maintenance schedules as shortly introduced above offers opportunities and improvements with respect to unscheduled maintenance. Research done in the field of deferring or rescheduling maintenance tasks will be presented in Section 3.2.

Maintenance Steering Group Approaches
In relation to the history of airline maintenance strategies as stated above multiple new approaches had been published by the Maintenance Steering Group in order to support maintenance decisions: MSG-1, MSG-2 and MSG-3.
MSG-1 and MSG-2 proposed an bottom-up approach which was an expensive solution because it is focused on components and parts instead of the performance of the overall system. Therefore the MSG-3 approach was developed as a top-down approach including the consequences and economic effects of failures. This method is still used today. MSG-3 consists of two different levels of analysis, where the level one analysis focuses on determining the failure category and level two on determining the appropriate maintenance task.
A schematic overview of the level one analysis can be seen in Figure 3.3. Depending on whether the failure is evident or not and whether the failure has operational effects or not each failure is categorized into five categories. Each failure needs to be analysed using this test.

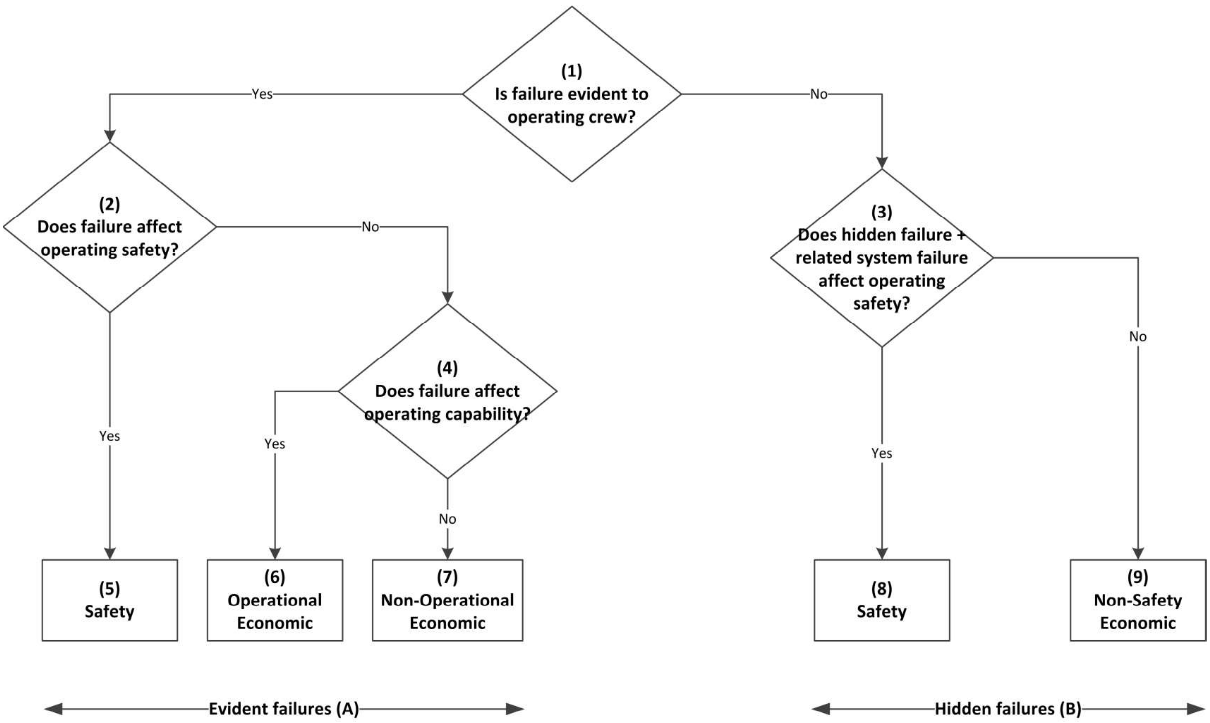


Figure 3.3: MSG-3 Level 1 Analysis [53]

The level two analysis differs depending on the result of the level one analysis. There are two flow charts, one for evident and one for hidden failures. Figure 3.4 shows the flow chart of the evident failures. In order to determine the appropriate maintenance actions safety related failures (category 5 and 8 in the level one analysis) need to answers all questions of the flow chart. Whereas non-safety related failures need to complete the flow chart until one positive (yes) answer is found. The flow chart of the hidden failures is very similar and can be found in literature [53]
As can be seen in the level two analysis a variety of tasks can be the result of the MSG-3 process. The tasks can then be combined with failure rate data in order to determine intervals until the next required maintenance action. Which can then again be used to improve the initial scheduled maintenance actions defined by regulations and requirements. This can then again be used to optimize maintenance schedules and to offer flexible maintenance solutions.

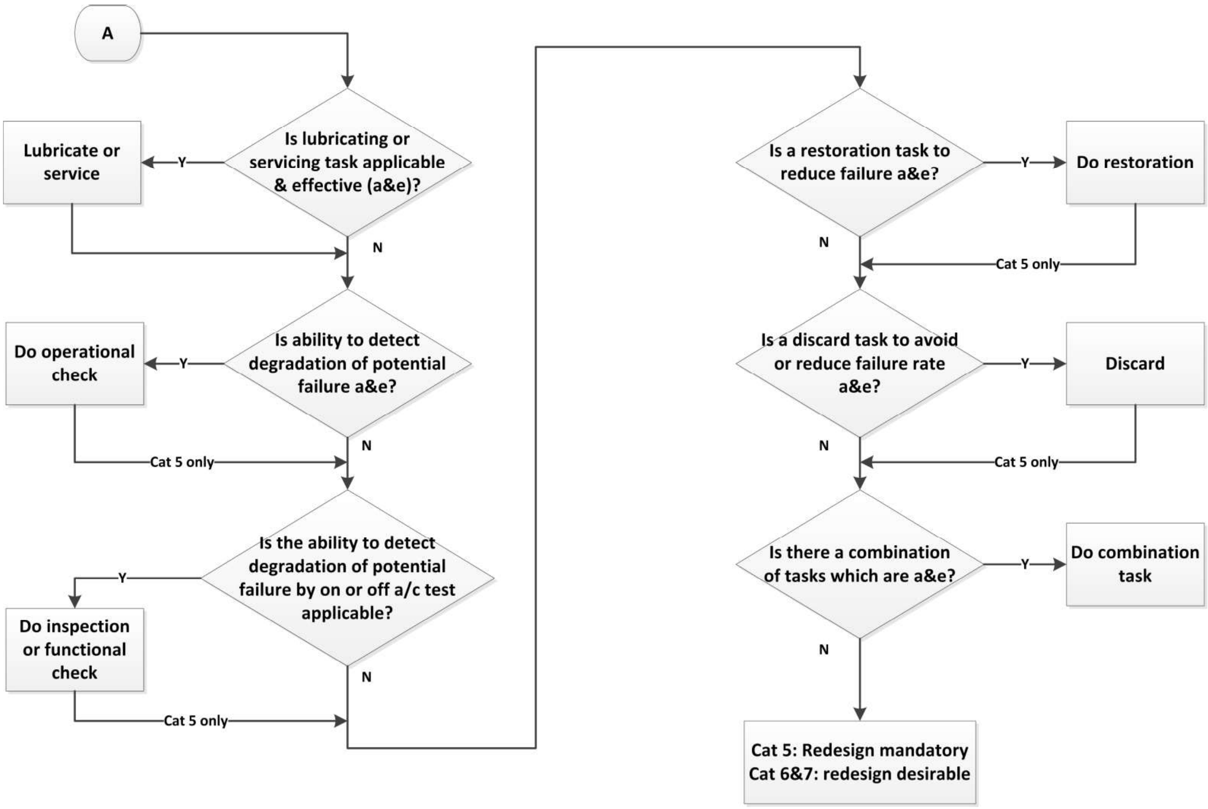


Figure 3.4: MSG-3 Level 2 Analysis - Evident Failures [53]

The latest version of the Maintenance steering group (MSG-3) is widely used today and in many cases already able to replace pre-described letter checks within airline maintenance, since it provides airlines with a complete decision process based on findings during inspections.

3.2. Current Research in Maintenance Scheduling

The previous Section described the general trends and forms of maintenance types and scheduling strategies available in the aviation industry. Those are however rather general and often not specific enough to choose the best possible maintenance action. A lot of theoretical work is done in the field of maintenance scheduling and task planning.

3.2.1. Line Maintenance

During line maintenance only the most urgent and short tasks can be executed. If unexpected problems occur it can happen that the aircraft needs to be delayed. Research has been done in the field of

short-term line maintenance planning by Papakostas et al. [54]. The aim of the research is to maximize fleet operability and minimize maintenance cost by developing a decision support tool to defer maintenance actions if required and possible. Constraints are formed in relation with respective costs, remaining useful life, risk and flight delay.
Further improvements to reduce delaying the aircraft due to longer line maintenance activities is proposed by Muchiri and Smit [55]. The proposed model is able to group tasks into appropriate packages in order to minimize wear and tear due to opening and closing of panels but also in order to defer packages to more other maintenance opportunities if the planned line maintenance time would be exceeded.

3.2.2. Base Maintenance

General requirements about when letter checks as a form of base maintenance are needed are prescribed in regulations and the maintenance planning document. Those are however only general and therefore airlines and researchers focus on scheduling those activities in the existing operational schedule. Sriram and Haghani [56] developed a short term (7 day) maintenance allocation model focused on domestic flights for which the aircraft routing assignment is already done before maintenance can be scheduled. Planned A- and B-checks can then be scheduled according to existing maintenance slots at existing maintenance bases. The model is however able to change tail number allocations if needed.
Lately research has also been done in order to combine a minimization of aircraft downtime and maintenance cost by implementing a more flexible maintenance strategy less focused on rigid letter checks [57]. The proposed model considers tasks individually according to the MSG-3 approach and assumes that tasks are executable during all ground time opportunities and therefore do not require predefined letter checks.

3.3. Cost Factors in Airline Maintenance

A 2015 IATA (International Air Transport Association) study analysed 60 airlines as part of their 2014 report and presented that approximately 10% of airline cost elements consists of maintenance cost as can be seen in Figure 3.5. This large financial impact is one of the main reasons for the intensive work done in the fields of diagnostics, prognostics, maintenance schedules and scheduling and planning optimization in the scientific world. It is however very complex to firstly identify all cost factors and secondly implement them in an optimization approach. Before a choice about the cost factors for optimization can be made an overview of the overall airline maintenance cost is given in this section.

Boeing presented cost factors as the built-up of the total maintenance cost (TMC) of an airline as can be seen in Figure 3.6. The main distinction is made by them between direct and indirect maintenance cost.
Costs immediately linked to the maintenance action itself are classified as direct maintenance cost and as shown in Figure 3.6 are according to Boeing either part of the airframe or the powerplant. Direct maintenance cost (DMC) consist of labour and material expenses required to perform the actual maintenance action [58].

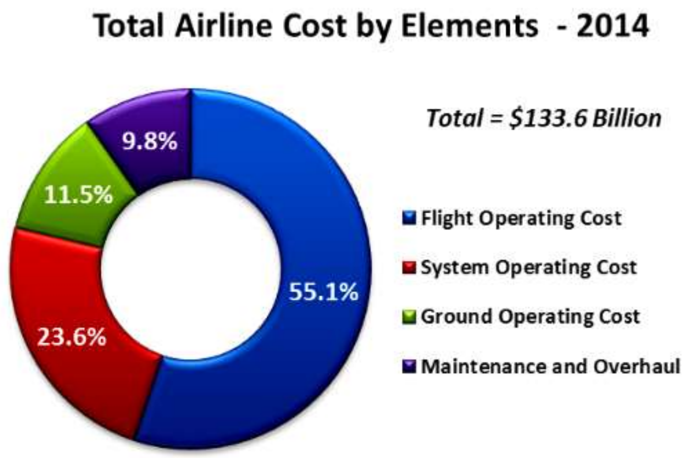


Figure 3.5: Total Airline Cost Elements in 2014 [59]

It is furthermore possible to classify the direct maintenance cost further than elaborated by Boeing. The main categories would then be [60]:

- Labour cost
- Repair cost
- Cost of lost utilization
- Risk cost

Even though some seem to be belonging to indirect maintenance cost, their amount is depending on the actual maintenance action which classifies them as direct maintenance cost. [60]

The labour costs include the direct work and man hours needed to repair the component but also possible time to get to location. This would happen if the technicians have longer travel time to the aircraft than usual or if special effort is needed to bring the new component to the aircraft.
A crucial aspect is of course also the actual repair cost, meaning the material cost of the maintenance action. The work done is not included in this part as it belongs to the labour cost and would otherwise be counted multiple times.
When looking at predictive or preventive maintenance action an important aspect to include is the lost utilization of the component. Wasting useful lifetime of a component by replacing it to early is a negative cost factor within direct maintenance cost. It is however almost impossible to accurately determine the lost utilization due to the fact that if the component is repaired or replaced one might never know the exact failure time in case the component would not have been replaced. In order to take this factor into account the actual lifetime can be compared to the budgeted life time of that component or also the day of replacement with the day with the highest failure probability.
Every time a strategy for a specific maintenance action is chosen a respective risk is related to that choice. Risk is mostly focused on delay of completion of the repair and therefore a possible increase in down-time. This delay risk then possibly results in increased labour cost of the maintenance technicians, increased fleet cost since the aircraft might not be usable on time and other (costly) solutions are required, increased crew cost if crew has a fixed salary, increased passenger cost if the delay is long enough that the passengers need to be compensated for and finally even cost factors down the operational line due to fleet and network effects.

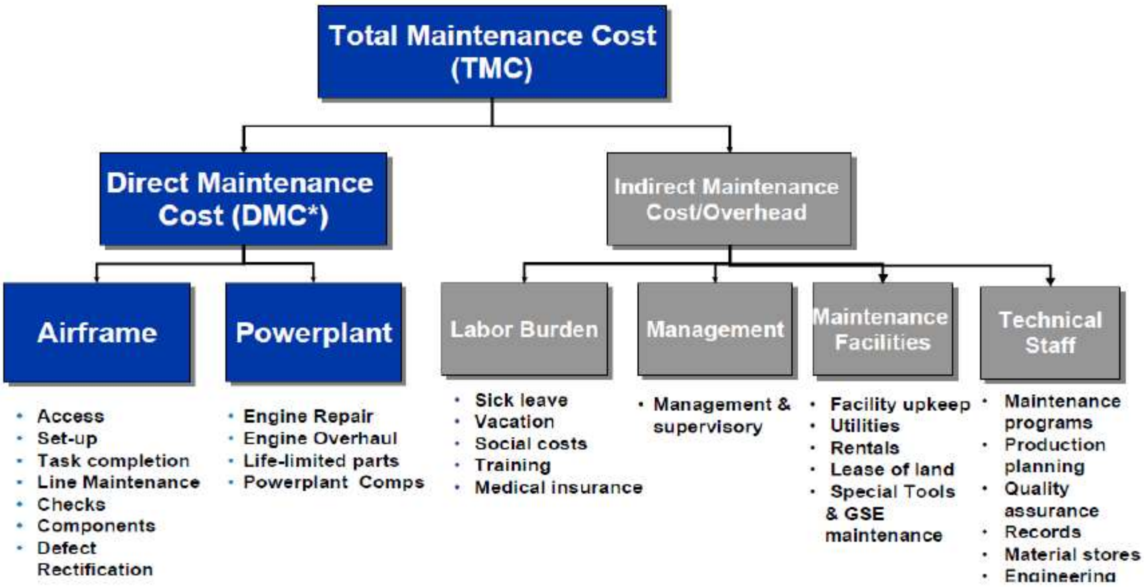


Figure 3.6: Total Maintenance Cost [61]

As can be seen in Figure 3.6 the other large part besides DMC are the indirect maintenance cost factors. Those expenses are unrelated to one specific maintenance action at a specific time and can be seen as fixed cost required to keep the maintenance company operational. It includes labour related costs which exist independent of the maintenance work done by the technicians including aspects like vacation and sick leave but also insurance and social costs. Another large part of the fixed cost are the facilities, tools but also planning, quality assurance and training of staff members. General maintenance actions provide further indirect maintenance expenses.

As was described above it is possible to differentiate between a large amount of different types of cost factors and cost definitions in the maintenance industry. Some cost factors like cost of lost utilization and risk cost are almost impossible to accurately determine for a real life situation. Other cost factors related to indirect cost are often left out during scheduling assignments due to fact that those factors are not directly reduced by changing a specific aspect of the maintenance schedule. Only large changes in the overall maintenance strategy would also have an effect on the managerial indirect cost factors.
Current maintenance scheduling research as described in Section 3.2 simplify cost largely in order to minimize the computational efforts. Papakostas et al. [54] uses a detailed maintenance cost per task per component which includes equipment-, labour- and overhead-rate, the maintenance time per component and component procurement costs. Whereas Muchiri and Smit [55] heavily simplify cost into just including human labour and neglecting parts or other overhead cost. Simply taking an average assumption of how expensive a certain type of maintenance check was done by Sriram and Haghani [56]. Senturk and Ozkol [57] developed a model to minimize downtime of the aircraft and allocated an average cost per downtime day.

3.4. Current Status of Real-life Airline Maintenance Operations

As stated above airline maintenance strategies changed tremendously from the first phases of manned flight up until now. Due to the fact that maintenance accounts for 10% of airline operation cost it is an important factor in order to decrease overall airline cost.
The main trend during the last years was to improve preventive and predictive maintenance schedules based on information about the failure behaviour of individual or more complex components.

Implementing and improving those scheduling procedures in a theoretical and scientific context as stated in Section 3.1, Section 3.2 and later on in Chapter 4 often offer good solutions to determine improved schedules but they are however often rather general and first and foremost theoretical. As explained above the main downside of many theoretical scheduling approaches is the highly simplified operational context by for example neglecting number of aspects or by oversimplifying real-life constraints. Therefore many airlines often do not actually apply those new schedules in detail. The main strategy in many cases is often still either manual trial and error iterations to find a feasible solution or very simple decision support tools.

This is done by initially defining a general maintenance schedule on a strategic long-term level based on for example MSG-3 rules and letter check regulations and includes mostly tasks and calender-, cycle- or flight hour-intervals. The initial schedule is described in an maintenance planning document (MPD). The schedule then needs to be transferred into an actual maintenance planning specific to the airline which incorporates the required tasks which were defined in the MPD. An important objective besides planning all tasks is also to spread out the work evenly such that technicians will be able to perform the tasks efficiently and in an allocated time frame.
Based on the operational schedule and the flight hours, cycles and days related to that operational flight schedule of the airline the engineering department of the airline will be able to assign specific task to maintenance opportunities. Planning also needs to take a number of availability constraints into account. Not just the aircraft itself needs to be available according to the operational schedule but the engineering department also needs to ensure that it has sufficient space in their maintenance facility, that enough man hours are available and that the required spare parts are available.
One of the important considerations of the maintenance planner when choosing tasks is the method of grouping as further explained also in Chapter 4. If it is possible to define overlaps either with respect to interval times, location in the aircraft, access point or task similarity it is most of the time smart to group those tasks together in a task package.

A large difference can be seen between the theoretical knowledge about good maintenance schedules and the actually applied maintenance schedules in airlines. More work is required in order to balance out this difference. Another important movement during the last few years (in the scientific field) but also most likely of the upcoming years was and is the predictive maintenance framework. As just stated before this is at airlines currently mostly done using manually decision processes. The main development will be the automatization of those decisions in a way to fully utilize components and operating schedules while minimizing cost. Chapter 4 will further elaborate on the progress in that field.

4
Maintenance Planning Optimization

There are a number of factors which need to be considered in airline maintenance schedules as it has been explained above in Chapter 3. The main challenge is however to optimize the planning of those maintenance activities. Sections 4.1 and 4.2 present the options of using prognostics information and multi-components in maintenance planning optimization. Finally an overview of the most used optimization algorithms is given in Section 4.3.

4.1. Prognostics in Maintenance Planning Optimization

The developments in the prognostics research as mentioned in Chapter 2 allow for new and improved maintenance schedules in the industry. Condition-based maintenance (CBM) policies propose a framework to incorporate prognostic advancements into maintenance scheduling to achieve higher system availability and cost reduction. A US study of 2003 predicted a 35 billion dollar per year cost reduction in the US if CBM would be fully utilized to minimize unexpected down times [62]. Unfortunately a more recent study found that companies actually observe a large difference between the potential of prognostics and the actual achieved benefits [63]. Apparently more research is needed to find solutions which actually offer a complete and easy to use application incorporating a variety of CBM aspects, which furthermore are linked to real-life airline operations.
Over the years a number of research papers have been published to propose a variety of implementations of prognostics information into maintenance schedules. There is not yet an overall agreed-upon optimal strategy to use prognostics and therefore most approaches focus either on different smaller specialized aspects which increases the difficulty to compare them directly [64, 65] or on the very broad point of view of prognostics in a maintenance schedule [66–69]. An overview of a variety of research approaches using prognostics information in a maintenance planning framework can be seen in Table 4.1. The table presents the operational context and scientific novelties of the studies including their main assumptions, type of prognostics and optimization used in order to optimize the objective function.

Applying prognostics but mainly focusing on another topic occurred in the research of Lei & Sandborn [64]. Most wind farms are managed using a power purchase agreement (PPA) which is a contract between an energy buyer and a wind farm operator. Before this method was proposed no researcher applied the agreed upon aspects of the PPA to a maintenance optimization process. It takes into account the minimum and maximum amount of energy sold, energy prices and price penalties in case of not adhering to points of the PPA. Therefore the objective is not to minimize maintenance cost but to maximize revenue according to the PPA based on a schedule using RUL information. The implemented RUL information was however not as sophisticated as explained in Section 2 and thus less meaningful.
Multiple authors furthermore worked on optimizing prognostic maintenance schedules while focusing on remote or distributed locations of the assets. Those proposals were usually applied to (offshore) wind turbines or railroad works. One of those studies combined a maintenance schedule

optimization with the Travelling Repairman Problem (TRP) and therefore incorporating travel time into the maintenance planning process [65]. During the overall process and the case study the focus however was mostly on minimizing the travel effects instead of applying failure probability prognostic information.

As stated above other research topics are rather broad when optimizing the maintenance schedule with respect to prognostics. One of the proposed implementations of prognostics into maintenance scheduling triggers the maintenance scheduling framework as soon as the estimated RUL is smaller than a defined lifetime margin [66]. The remaining useful life predictions are repeated until that value is below the defined lifetime margin after which the predictive maintenance scheduling occurs. Therefore the found maintenance optimization might not actually be optimal since not all possible combinations are evaluated. This procedure furthermore only focuses on one single component.
Most research strategies used actual RUL estimations (with different levels of accuracy) as an input for their optimization approach. Another option applied by Camci [67] is to use prognostics in the form of failure probabilities of multiple components to create an objective functions which minimizes risk (failure and maintenance risk). This is done by utilizing prognostics and reliabilities but also inventory and maintenance data. This study furthermore analyzed the difference between whether or not applying thresholds to the predicted failure probabilities and therefore suggesting that the solution of You and Meng [66] might not be optimal.
A study on maintenance scheduling using reliability and prognostics information heavily simplifies the operational schedule [68]. The operational context of the research is a military air force base with ten aircraft. The operational schedules of a military or a commercial operator are in most simplified optimizations rather similar due to the fact that most airlines simplify their maintenance operation also to one maintenance location. The main difference with respect to operations is therefore the fleet size and the capacity. Another difference between the research and actual commercial aircraft maintenance is that the proposed algorithm simplifies maintenance cost to a maintenance capability and therefore maximum number of aircraft which can be maintained. An optimization approach in a commercial context would need to take more cost factors into account in order to determine the best maintenance opportunity.
Unlike most other approaches took Zhang et al. [69] imperfect maintenance actions into account but the actual planning optimization based on RUL estimates was strongly simplified. When the RUL estimate at inspection is smaller than the time interval until the next inspection then the maintenance action is executed immediately. The actual maintenance time stamp is therefore not actively planned in this planning approach.

Due to the struggles of computing exact RUL predictions as explained in Chapter 2 the implementation is currently still a challenge. An optimization based on information with a high level of inaccuracy results in less meaningful decision support tools. It is presented above that there are a number of studies trying to implement prognostics but often deliver results with many assumptions related to the operational context of the system. Therefore most of the results are currently still mainly scientific and not yet ready to be implemented in real life airline operations.

Table 4.1: Overview of Research done with respect to Prognostics Implementation in Maintenance Planning Optimization

Ref.	Year	Operational Context	Novelty	Assumptions	Prognostics	Objective Function	Optimization Approach
[64]	2018	Windfarms	including PPA in maintenance optimization	strong grouping due to remote location; simplified RUL predictions based on wind, rotor rotations and Weibull distributions	using RUL	max net revenue	Real Option Analysis
[65]	2015	distributed assets like offshore and windfarms and railway switches	including travel time in the scheduling optimization	one maintenance team starting and ending at base; maintenance is always possible (if hours available per day)	using failure probability prognostics	min cost (failure, maintenance, travel cost)	Genetic Algorithm
[66]	2012		using RUL to minimize wasted component life	maintenance can be performed immediately when required; duration and cost are not directly linked to the system at stake	triggered when RUL smaller than defined lifetime margin	min cumulative system risk (expressed in cost)	
[67]	2009		comparison between setting thresholds to failure probabilities or not		failure probabilities (with and without thresholds)	min risk (failure and maintenance risk)	Genetic Algorithm
[68]	2016	Airforce operation		prognostics implemented as average remaining flying hours and a uniform distribution; maintenance cost implemented as maintenance capacity	RUL and related uniform distribution	min the maximum difference between used and available maintenance slots	MIP
[69]	2018		implementing imperfect maintenance; continuous degradation and instantaneous degradation due to shocks	set inspection intervals; estimated degradation process to obtain RUL; immediate maintenance action if RUL smaller than interval until next inspection	RUL based on threshold of estimated degradation status	min maintenance cost	Genetic Algorithm

4.2. Multi-Components in Maintenance Planning Optimization

The optimization of condition based maintenance is an important topic in the industry. Unfortunately a large part of those studies analyse CBM optimizations of single components. A few researchers however put a special focus on multi-components in the optimization of CBM, however most of those studies include other specializations in the research as well or deliver a very specific solution to a particular multi-component system. A variety of scientific contributions in the field of applying multi-components in maintenance scheduling have been analysed and are presented in Table 4.2.

As explained in Section 2.2 many multi-components show a positive economic dependence and therefore most of the in literature existing multi-component CBM optimizations are based on PED to minimize maintenance cost [70–75]. This has the advantage that those solutions can often be adapted easily in order to fit other systems as well. Another strategy is to closely look at the physical and structural interdependencies as done in [76–79]. This however limits the applications to that specific system instead of offering a general solution for a variety of multi-component structures [48]. In recent years more and more approaches included importance measures as part of their multi-component maintenance planning optimization. Component importance is used to prioritize components in order to select the most crucial ones in a system to focus on during an optimization [80]. Examples of applying those importance measure to maintenance optimization can be found in [76, 79].

An often considered approach to implement a multi-component perspective is the method of grouping. Generally speaking this method works based on maintaining one component when another is maintained. Literature considers three different categories of grouping. Long-term (static) grouping, medium-term (dynamic grouping) and short-term (opportunistic) grouping. The main strategy of grouping is to minimize set-up cost under the possible down side of increasing down-time cost. [81]
Organising planned preventive maintenance activities in a way that the tasks can be done at the same instance, for example during a specific letter check is a form of static grouping which is done during the strategical maintenance planning phase. In most cases this static grouping is focused on inspections or servicing activities.
Dynamic medium-term grouping usually considers which inspections or services can be executed when a planned preventive or corrective repair is performed. A number of studies applying dynamic grouping [70, 72, 77, 78, 80] can be seen in Table 4.2.
A further step is taken in the short-term opportunistic grouping [72, 74–76, 81–84]. This type of grouping even considers unplanned repair or replacement actions and which planned inspections or repairs could be combined with it. This grouping strategy is most likely to be very important in the proposed application of prognostics in multi-unit maintenance schedules due to the comparatively short horizon of good RUL predictions.

The collection of presented literature below show a number of different novelties included in the optimization process, but also a variety of assumptions made in order to achieve the optimization goals. A number of differences are elaborated on in the next paragraphs and information about which references applied those aspects can be seen in Table 4.2.
One of the large difference between the studies is whether or not corrective maintenance (CM) is considered a possibility during maintenance planning. Approaches neglecting CM assume that their prognostic knowledge is perfect thus that all failures can be predicted, which is not always realistic or close to actual operations. But the risk of allowing CM is that it minimizes the possible optimization field since the time slot of CM itself can not be optimized.
Another aspect is whether and how prognostics are implemented. The set of presented literature

is focused on implementing multi-components and not prognostics (which is shown in Section 4.1) but prognostics information or at least degradation information is required in order to truly optimize preventive and predictive maintenance. Most of the presented solutions incorporate this information in form of a Gamma process to model the approximated degradation process. Applying actual prognostics would add additional value to real life meaningfulness of the optimization.
Many approaches furthermore assume instantaneous inspections and repair which are immediately possible whenever required. The advantage of this assumption is that it is easier to implement and also that the approach is therefore applicable to multiple situations and systems. The disadvantage however is lack of correlation with real-life maintenance. A specific action can only be planned when there is enough time during that maintenance slot.
Finally a difference between maintenance performance can be observed. A number of approaches assumes perfect maintenance which highly simplifies the model whereas imperfect maintenance would often describe the real life application more accurately.

It is important to apply maintenance scheduling optimizations not just to simple single components but also to larger systems consisting of multiple components and units. The current status of the research in this field as presented above however shows that that degradation states, maintenance duration and operability constraints are heavily simplified in the studies available and the optimizations are therefore not ready to be used in real life situations at airlines.

Table 4.2: Overview of Research done with respect to Multi-Component Maintenance Planning Optimization

Ref.	Year	Novelty	Assumptions	Component Dependence	Grouping Strategy	Objective Function	Optimization Approach
[70]	2011	continuous updating of schedule	Gamma degradation process; instantaneous repair; CM possible	PED	dynamic	min maintenance cost	Heuristic algorithm
[71]	2005	opportunistic CBM	two unit system; instantaneous repair; maintenance available whenever needed	PED	oppor-tunistic	min long run maintenance cost	
[72]	2012	variety of dependencies, including imperfect maintenance	spare parts always available	PED and structural	dynamic and oppor-tunistic	min long term maintenance cost	local opti-mization
[73]	2012		linearly increasing failure rate; predefined inspection slots	PED	dynamic	min total maintenance cost	GA
[74]	2010	multi component CBM	predefined inspection intervals, instantaneous repair, independent component degradation	PED	oppor-tunistic	min maintenance cost	local opti-mization
[75]	2013		multiple identical components; instantaneous repair, perfect maintenance	PED	oppor-tunistic	max revenue of the system	MDP and heuristic algorithm
[76]	2014	logistic constraints in grouping	Gamma degradation process; instantaneous, perfect inspections and repair; maintenance possible only at predefined inspection times	PED and structural	oppor-tunistic	min maintenance cost	Monte Carlo Simulation
[77]	2015	decision support on two levels (component and system level)	Gamma degradation process; perfect maintenance	PED and structural	dynamic	min long-run maintenance cost	Monte Carlo Simulation
[78]	2014	using positive and negative economic dependence on one system	preventive and corrective maintenance; instantaneous repair; immediate repair when needed	PED and NED	dynamic	min maintenance cost	GA
[80]	2016	applying component maintenance priority as an importance measure	no CM; components are statistically independent	PED and structural	dynamic and oppor-tunistic	min maintenance cost	local opti-mization
[81]	2016	preventive maintenance duration is taken into account	CM possible; instantaneous repair	PED	oppor-tunistic	min maintenance cost; max system availability	PSO Algorithm
[82]	2017	local prognostic control at component level and global optimal problem at system level			oppor-tunistic	min maintenance cost	SBM
[83]	2013	including environmental changes	predefined sets of health states and environmental conditions	PED	oppor-tunistic	min long run maintenance cost	MDP and policy iteration method
[84]	2012	DBN-HAZOP model to predict degradation on component level	perfect repair; CM possible	stochastic, structural and economic	oppor-tunistic	min maintenance cost	local opti-mization

4.3. Optimization Algorithms

There are many different strategies to determine an optimal maintenance schedule as presented in the previous sections of this chapter. This section will highlight some of the most used optimization algorithms in this field.
Even though there are many different approaches to optimize the maintenance schedules it can be seen that often the same two optimization algorithms are used. Table 4.3 presents a number of examples of genetic algorithms (GA) and Markov decision processes (MDP) applied in the field.

4.3.1. Genetic Algorithm

The genetic algorithm is a form of meta-heuristic algorithms and it is based on the analogy to Darwin's theory of evolution. One important characteristic of meta-heuristic algorithms and therefore also of genetic algorithms is that they are usually able to determine a very good and feasible solution in a short computational time. However it is not possible to ensure that the found solution is the overall global optimum for that optimization problem. [85]

Initialization is the first step of the GA, during which an initial population is determined or created. After which the fitness with respect to the objective function of each member is evaluated.
The second step is the first iteration. As part of Darwin's theory it can be said that members with a higher fitness are more likely to create off-springs. Therefore a random selection of the population with a bias towards members with higher fitness is created. That selection is then used as parents in the first iteration. Each pair creates two children and each child inherits a random selection of features from both parents. The fitness of the new generation is again determined which is then the start of the next iteration round.
It is furthermore possible to implement a mutation rate into the iteration process. This means that the children would have some features which were not existing within their parents. This and the amount of population size is an important tuning variable of the optimization process since this can either increase or decrease the computational time depending on the amount of mutation and population.
Creating a stopping rule is the final step of the genetic algorithm. It is possible to stop the algorithm either after a certain amount of iterations or computational time or after a specified amount of iterations without any changes in the members with the best fitness.

One of the advantages of GA is that it can be implemented even if the optimization problem requires a non-linear objective function. Literature presents a number of maintenance optimization problems applying a genetic algorithm as in [65, 67, 73, 86, 87].
The examples in the table below show that it is possible to apply a GA to a variety of tasks within the optimization process. Depending on the objective of the approach it can be used to determine the best maintenance slots but also to find the optimal sequence or group of components for a specific slot.
When implementing the genetic algorithm as explained above a number of decisions about the GA operators need to be made. Different studies chose to either use existing, calculated or approximated operators in order to define the population size, mutation rate, crossover and stopping rule.

4.3.2. Markov Decision Process

Markov chains are an important aspect of stochastic processes and can be applied to a variety of real life problems. In order to formulate the problem as a Markov Decision Process (MDP) it is crucial that a clear set of states can be defined. Another characteristic is that any Markov process is a forward-looking process, therefore any event in history does not effect the decision in the future. The decision is purely based on the state at the time of inspection. [85]

The MDP can be characterized by a number of steps which need to be followed. Initially a set of states and their respective conditions and a set of actions are required to be determined. After each transition the state can be observed and a decision about the action can be made. Not all actions are possible for all states, therefore those dependencies are required to be established in the definition of sets in the beginning. Based on the observed state and chosen action a cost factor for that specific combination and a transition probability for the next transition can be determined. The objective is then to find the order of actions resulting in the lowest expected average cost per unit time depending on the chosen actions at various states.

Maintenance schedule optimizations provide a suitable real life situation for a MDP. The degradation levels are easily translated into states as done in [75, 83, 88, 89]. A MDP solution however easily gets very large and complicated if the number of states is high. In case of many states or interdependencies which needs to be taken into account.
Some of the examples use the MDP in order to determine at all decision points whether or not a maintenance action should be performed or not. Some even determine which kind of maintenance task(s) would be optimal for the situation at hand. It is however also possible to use a Markov decision process to model the degradation process of individual components as shown in [75].
For all MDPs it is crucial to define a number of sets of for example health states, degradation states, transition rates and transition conditions.

Table 4.3: Overview of Optimization Approaches in Literature

Ref.	Year	Optimization Approach	Assignment of the Algorithm	GA operator/ MDP characteristics
[65]	2015	GA	assigning components to a time slot and ordering the components per time slot	existing operators (scattered crossover and Gaussian mutation)
[67]	2009	GA	evaluating risk	existing operators (scattered crossover and Gaussian mutation)
[73]	2012	GA	identifying the important components and optimize the maintenance periods	RSM is used to determine operators
[86]	2012	GA	determine best sequence of maintenance actions	one point crossover process and set mutation rate
[87]	2000	GA	determining maintenance time and component combination per repair	standard operators (single random splice crossover and set mutation rate)
[83]	2013	MDP	determine at which decision points a maintenance action should be performed	set of health states; set of environmental condition states; set transition rates and conditions
[88]	2005	MDP	determine next inspection time and determine what kind of maintenance action (major, minor, no repair)	calculated transition probabilities between states
[89]	2005	MDP	determine whether or not to perform CM and PM and if no maintenance how much the system should produce	set of system states, time periods, failure probabilities and completion probabilities
[75]	2013	MDP	modelling degradation process of independent components	set of component health states

III

Elaborations of Thesis Work

5
Prognostics as an Input to the Model

This chapter is an addition to Part I and further elaborates on the prognostics used in the case studies. This research uses classification component prognostics developed for the bleed air system of a wide-body fleet of a European airline [90]. To implement the prognostics, it is first explained in Section 5.1 which prognostics information from the model is used in the maintenance scheduling model. After which an approximation of the failure function is presented in Section 5.2. Then the implementation of the prognostics in the model is presented in Section 5.3.

5.1. Prognostics information obtained from the tool

The maintenance scheduling approach combines component prognostics and available maintenance slots for a certain time frame to determine when to repair the component. The chosen component prognostics model is unfortunately not available to obtain actual prognostics information and therefore the presented results of the research are used as an input to the maintenance scheduling model.

One conclusion from the component prognostics model is that 63% of the predictions made are ’no’ predictions and therefore only 37% are ’yes’ predictions.
It is furthermore also possible to determine the positive and negative predictive values.

- Positive Predictive Value (PPV) = Prediction = $\frac{TP}{TP+FP}$ = 0.31
- Negative Predictive Value (NPV) = $\frac{TN}{TN+FN}$ = 0.86

Where TN is the number of true negatives associated with the prognostic, FN is the number of false negatives associated with the prognostics, TP is the number of true positives associated with the prognostic and FP is the number of true negatives associated with the prognostics.

Meaning that only 31% of the yes predictions correctly predict a flight deck error (FDE) in the upcoming ten days and that 86% of the no predictions correctly predict that no FDE occurs in the next ten days.

The classification component prognostics model is intended to be run on each consecutive day to determine on every day a prediction for the upcoming 10 days. If one is at a specific day the current prediction for the next 10 days is known and also the previous predictions would be available for their respective time frames as shown in Figure 5.1.

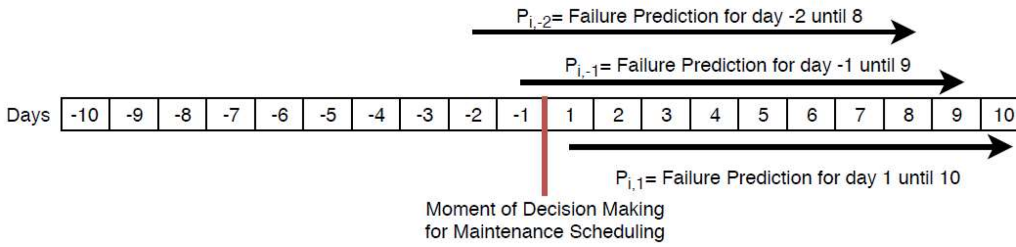


Figure 5.1: Overview of the prognostics

5.2. Failure Function Approximation

The classification component prognostics model does not provide any failure probability information and therefore an approximation needs to be found if one wishes to include a certain increase in failure probabilities.

Since 94% of all components show a constant hazard function [52] a failure distribution which leads to a constant hazard function is most likely to be correct. The exponential distribution is the only continuous distribution with a constant hazard function and therefore an exponential function is assumed for future steps.

The rate parameter λ describes the average failure rate as can be seen in equation 5.1. Thus on average once in 354 days the bleed air system fails on the examined fleet used in this prognostics model implementation.

$$\lambda = \frac{1}{354} \tag{5.1}$$

The failure function can then be calculated as shown in equation 5.2 and the function can also be seen in Figure 5.2.

$$F(x) = 1 - exp(-\lambda x) \tag{5.2}$$

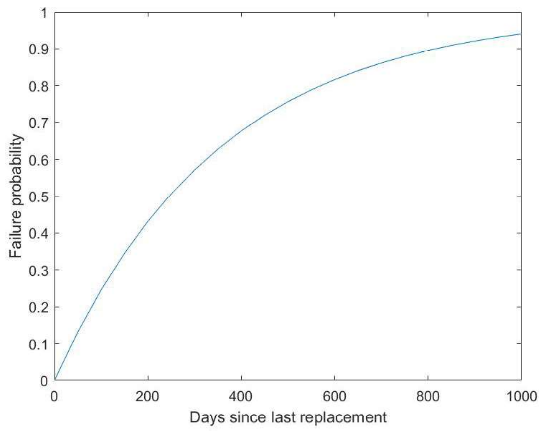


Figure 5.2: Approximated Failure Function of the Bleed Air System

Knowing the time since last repair or replacement it is, therefore, possible to approximate the failure probability at a specific day (see equation 5.3.

$$F(t_r) = 1 - exp(-\lambda t_r) \tag{5.3}$$

5.3. Implementation of the prognostics information in the MCTS

To use as much of the actual results of the component prognostics the failure function and the accuracy values of the prognostics are used in the maintenance scheduling model in the form of biases and punishments as shown in Table 5.1

Table 5.1: Implementation of prognostics information

	Implemented in MCTS
Failure Function	- As a bias when randomly creating predictions
PPV and NPV	- Choosing biased random simulated actions related to predictions - As punishments

Implemented as a form of punishments means for example that a decision not to repair receives a high punishment value if there was a yes prediction for failure with high accuracy. How exactly this is done is described later onward in the punishment section of this model.

How the predictions are implemented is a complicated topic due to the fact that actual predictions are not available. If the maintenance scheduling model would, however, be applied in the real situation where the classification component prognostics model would also be available the situation at hand would be as presented in Figure 5.1. Since those predictions are not known they are created randomly with a bias related to the amount of yes and no predictions in the confusion matrix.
The maintenance scheduling model will also use simulated predictions of future days which would not be available at a specific day when starting the optimization process. Using those simulated predictions which would not be known yet are the main advantage of the chosen modeling approach. The chosen action for a specific day can be related not just to the information from the past but also to possible future predictions.
An overview of the three types of predictions, where they come from, what they mean and how they are used can be seen in Table 5.2.

Table 5.2: Types of predictions and their creation and usage in the MCTS

	Made on which days?	**Made for which days?**	**Creation in MCTS**	**Usage in MCTS**
Current prediction	day 1	day 1 - day 10	Randomly generated with bias failure function	Punishment 2 (prediction is compared with simulated action)
Previous predictions	previous L_p days	each time the upcoming 10 days as shown in Figure 5.1	Randomly generated with bias failure function	Punishment 3 (includes 'known' predictions of previous days to enforce a decision towards repair or no repair based on the amount of Yes and No predictions in the last L_p days)
Future predictions	upcoming L_d days (not known yet on current day)	each time the upcoming 10 days as shown in Figure 5.1	Randomly generated with bias failure function	Punishment 2 (Simulated predictions in the future are compared with simulated actions)

6
Monte-Carlo tree search Model Description

This chapter again further elaborates on the Monte-Carlo tree search model presented in Part I. The Monte-Carlo tree search method is especially useful when handling simplified data similar to our approximated classification component prognostics input. The MCTS is also able to incorporate future predictions which are not yet known at initialization in the form of possible future situations. This section will first elaborate on the tree structure and the tree characteristics of the Monte-Carlo tree search. After which the simulation is described. Then the punishment structure is presented to evaluate the simulations. The main model then is extended by two model extensions. At the end, a description of all steps of the Monte-Carlo tree search is shown.

6.1. Tree Structure and Tree Characteristics

6.1.1. Layout of the Tree

The Monte-Carlo tree search (MCTS) considers a fleet of I aircraft and a scheduling horizon of L_d days into the future and takes the prior L_p days into account, where L_p is defined as a negative number ($L_p < 0$), as can be seen in Figure 6.1

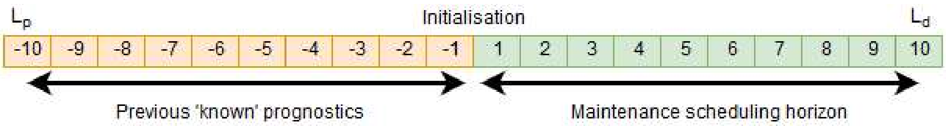


Figure 6.1: Time frame of the MCTS

The tree will therefore have L_d layers where each layer L_j, $1 \leq j \leq L_d$, has k_j nodes, $1 \leq k_j \leq n^j_{child}$. Where n_{child} is the maximum amount of possible child nodes and n^j_{child} the maximum amount of possible nodes on layer L_j.

Each layer L_j, which is the j^{th} layer of the tree, includes the following information valid for all $i \in I$ and $1 \leq j \leq L_d$:

$$L_j : (m_{1,j}, m_{2,j}, ..., m_{i,j}; P_{1,j}, P_{2,j}, ..., P_{i,j}) \tag{6.1}$$

Where $m_{i,j}$ is the maintenance opportunity of aircraft i on layer L_j and $P_{i,j}$ is the prediction to fail for aircraft i on layer L_j.

And each node n_{j,k_j}, which is the k^{th} node on layer L_j, includes the following information valid for all $i \in I$, $1 \leq j \leq L_d$ and $1 \leq k_j \leq n^j_{child}$:

$$n_{j,k_j} : (a_{1,j,k_j}, a_{2,j,k_j}, ..., a_{i,j,k_j}; S_{1,j,k_j}, S_{2,j,k_j}, ..., S_{i,j,k_j}; F_{j,k_j}; R_{j,k_j}; x_{j,k_j}; A_{j,k_j}; UCB1_{j,k_j}) \tag{6.2}$$

Where a_{i,j,k_j} is the action taken for aircraft i on node n_{j,k_j}, S_{i,j,k_j} the status of aircraft i (whether or not it has been repaired up until) on node n_{j,k_j}, F_{j,k_j} the infeasability flag of n_{j,k_j}, R_{j,k_j} the random rollout value of node n_{j,k_j}, x_{j,k_j} the counter of how often node n_{j,k_j} has been included in an iteration, A_{j,k_j} the award value of node n_{j,k_j} and $UCB1_{j,k_j}$ is the UCB1 decision value of n_{j,k_j}.

There is furthermore a root node n_0 on the 0^{th} layer of the tree, which is equipped with a counter of the root node x_0 and the Award value of the root node A_0. Their computations are similar to the counters and award values and will be explained below.

$$n_0 : (x_0; A_0) \tag{6.3}$$

All tree information and tree statistics stored in layers L_j and nodes n_{j,k_j} introduced above will be further elaborated on below. Figures 6.2 and 6.3 show an example of how the layers L_j and the nodes n_{j,k_j} are implemented in the tree.

6.1.2. Example Tree

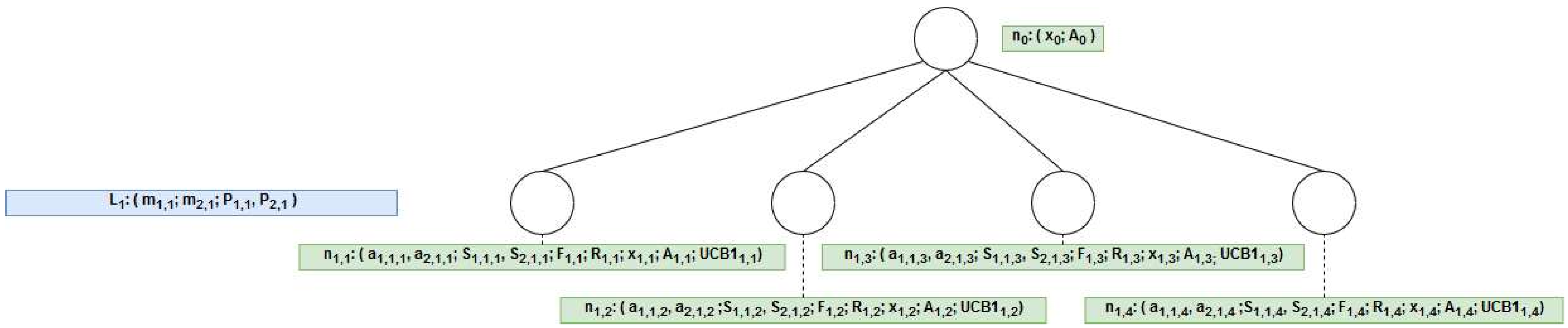


Figure 6.2: MCTS after fourth iteration (with variables)

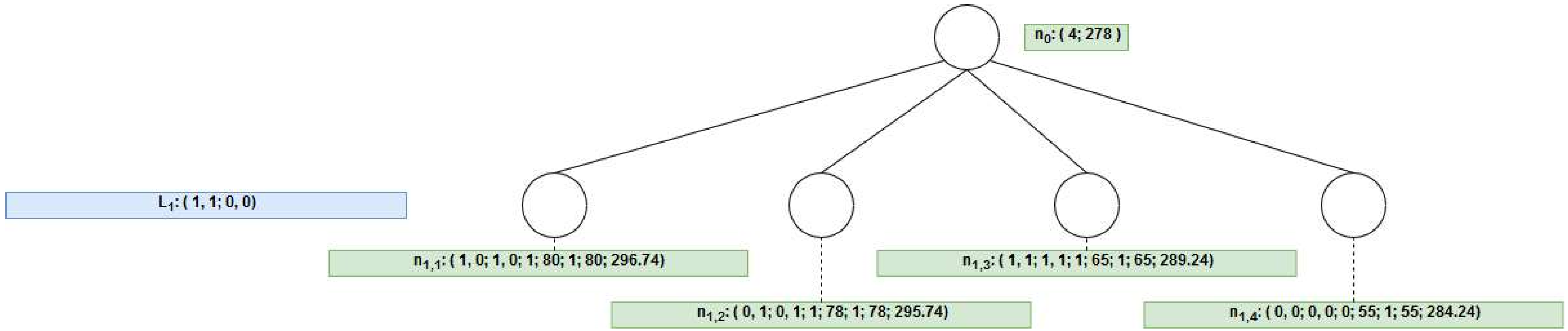


Figure 6.3: MCTS after fourth iteration (with sample values)

6.1.3. Node Creation and Numbering

Each layer L_j consists of multiple nodes n_{j,k_j}. Each node n_{j,k_j} is identified by an index k_j. Which index k_j is associated with the node is dependent on the index k_{j-1} of the parent node, the layer L_j on which the child node is initialized and the amount of aircraft I of the operational context.

$$k_j(k_{j-1}, j, I) \in \{(k_{j-1} - 1) \cdot n_{child} + 1; k_{j-1} \cdot n_{child}\} \tag{6.4}$$

As can be seen above, the possible values of k_j range from $(k_{j-1} - 1) \cdot n_{child} + 1$ until $k_{j-1} \cdot n_{child}$ and are therefore directly linked to the maximum amount of possible child nodes n_{child}. A few examples of the numbering scheme can be found in Figures 6.4, 6.5 and 6.6.

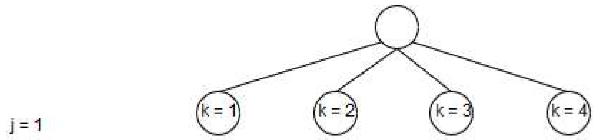


Figure 6.4: Sample node creation and numbering Part 1

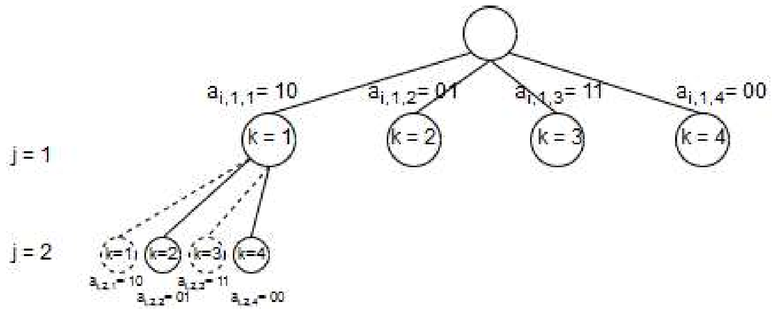


Figure 6.5: Sample node creation and numbering Part 2

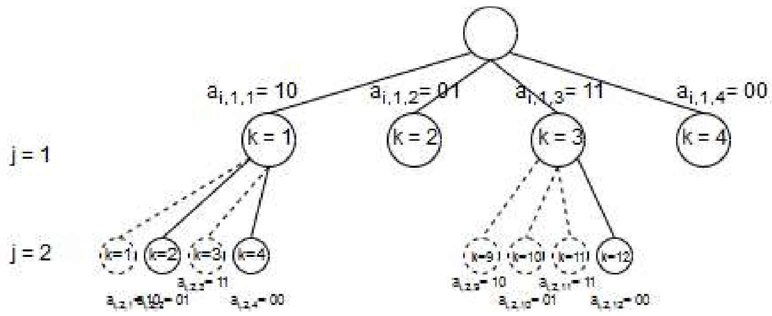


Figure 6.6: Sample node creation and numbering Part 3

Parent Node Definition

Node $n_{j-1,k_{j-1}}$ is the parent of nodes n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$. The index k_{j-1} of the parent node $n_{j-1,k_{j-1}}$ can be computed as $k_{j-1}(I, k_j) = \left\lceil \frac{k_j}{n_{child}} \right\rceil$. And therefore the parent node is presented as $n_{j-1,\left\lceil \frac{k_j}{n_{child}} \right\rceil}$.

6.1.4. Available Maintenance Slots

The maintenance slots $m_{i,j}$ are defined for each layer L_j, $1 \leq j \leq L_d$ and each aircraft i, $i \in I$.

$$m_{i,j} \in \{0,1,2\} \tag{6.5}$$

$$m_{i,j} = \begin{cases} 2 & \text{if AC } i \text{ has a specific maintenance slot for its tail number on day } j \\ 1 & \text{if AC } i \text{ has a generally available maintenance slot on day } j \text{ the type of AC } i \\ 0 & \text{if AC } i \text{ has no maintenance slot on day } j \end{cases} \tag{6.6}$$

At day 0 the values for $m_{i,j}$ are generated in advance and are stored in the respective layers L_j.

6.1.5. Action and State

The action a_{i,j,k_j} and the state S_{i,j,k_j} are defined for each aircraft i, $i \in I$, on node n_{j,k_j}, with $1 \le k_j \le n^j_{child}$ and $1 \le j \le L_d$, where n^j_{child} is the maximum amount of possible nodes on layer L_j.

$$a_{i,j,k_j} \in \{0,1\} \tag{6.7}$$

$$a_{i,j,k_j} = \begin{cases} 1 & \text{if at node } n_{j,k_j} \text{ it is chosen to repair AC } i \text{ on day } j \\ 0 & \text{if at node } n_{j,k_j} \text{ it is chosen not to repair AC } i \text{ on day } j \end{cases} \tag{6.8}$$

and

$$S_{i,j,k} \in \{0,1\} \tag{6.9}$$

$$S_{i,j,k_j} = \begin{cases} 1 & \text{if at node } n_{j,k_j} \text{ AC } i \text{ has been repaired between day 0 and day } j \\ 0 & \text{if at node } n_{j,k_j} \text{ AC } i \text{ has not been repaired between day 0 and day } j \end{cases} \tag{6.10}$$

Which actions a_{i,j,k_j} are possible depends on the state $S_{i,j-1,k_{j-1}}$ of the previous (parent) node $n_{j-1,k_{j-1}}$ and the maintenance opportunity $m_{i,j}$ at layer L_j.

$$a_{i,j,k_j}(S_{i,j-1,k_{j-1}}, m_{i,j}) = \begin{cases} 0 \text{ or } 1 & \text{if } S_{i,j-1,k_{j-1}} = 0 \textbf{ and } m_{i,j} = 1 \\ 0 & \text{if } S_{i,j-1,k_{j-1}} = 1 \\ 0 & \text{if } m_{i,j} = 0 \end{cases} \tag{6.11}$$

Those three options are shown below in Figure 6.7.

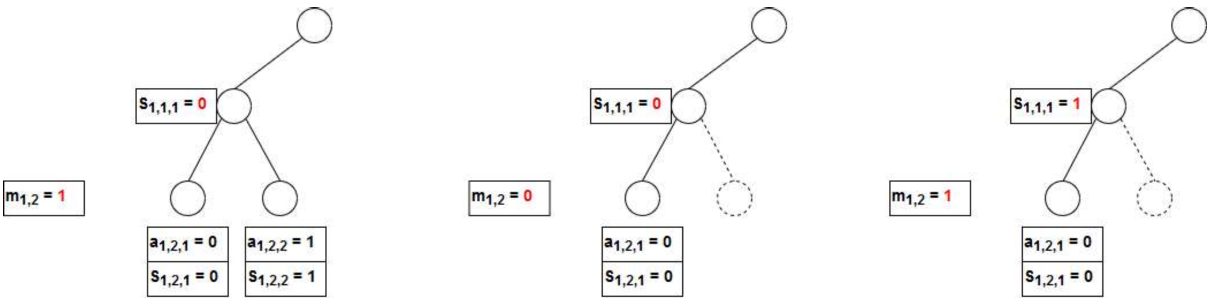


Figure 6.7: Sample Possible Actions depending on State and maintenance opportunity

The possible actions will furthermore be constrained by the maximum amount of possible repairs at the same time M_j. Therefore for a given layer L_j, $1 \le j \le L_d$, and a given node n_{j,k_j}, $1 \le k_j \le n^j_{child}$, on that layer, the following needs to be true:

$$\sum_{i=1}^{I} a_{i,j,k_j} \leq M_j \tag{6.12}$$

Similarly the state S_{i,j,k_j} of aircraft i at node n_{j,k_j} depends on the state of the previous visited node $n_{j-1,k_{j-1}}$ and the action a_{i,j,k_j}.

$$S_{i,j,k_j}\left(S_{i,j-1,k_{j-1}}, a_{i,j,k_j}\right) = min\left\{S_{i,j-1,k_{j-1}} + a_{i,j,k_j}, 1\right\} \tag{6.13}$$

It is furthermore important to specify the order of actions $a_{i,j,k}$ associated with each node $n_{j,k}$ such that the above-mentioned node numbering scheme works. This is ensured by creating a fixed order of actions on day 0. The order is stored in a matrix AO with a size of (I x n_{child}), where n_{child} is the maximum amount of possible child nodes.

The matrix AO shows the different aircraft i, $i \in I$ in the columns and each row presents the actions for the different nodes k_j, $1 \leq k_j \leq n_{child}$.

$$AO = \begin{bmatrix} a_{1,1,1} & a_{2,1,1} & \ldots & a_{I,1,1} \\ a_{1,1,2} & a_{2,1,2} & \ldots & a_{I,1,2} \\ a_{1,1,3} & a_{2,1,3} & \ldots & a_{I,1,3} \\ a_{1,1,4} & a_{2,1,4} & \ldots & a_{I,1,4} \\ \ldots & \ldots & \ldots & \ldots. \\ a_{1,1,n_{child}} & a_{2,1,n_{child}} & \ldots & a_{I,1,n_{child}} \end{bmatrix} \tag{6.14}$$

Meaning that the pattern of actions a_{i,j,k_j} attributed to node n_{j,k_j} for all aircraft i, $i \in I$ and layers j, $1 \leq j \leq L_d$ is repeated as followed:

$$a_{i,j,b \cdot n_{child}+k} = a_{i,1,k} \text{ with } b = 1,2,3,4,5,...2^{(j-1)\cdot I} \tag{6.15}$$

It can indeed be seen in Figure 6.6 that $a_{1,1,1} = a_{1,2,1} = a_{1,2,9} = 1$ and that $a_{2,1,4} = a_{2,2,4} = a_{2,2,12} = 0$.

6.1.6. Prediction

The MCTS considers a scheduling horizon of L_d days into the future and takes L_p days of the past into account. As explained in Chapter 5, the predictions are randomly created to represent the actual prediction output of the prognostics tool. A yes/no prediction is made for each aircraft i, $i \in I$, and for each layer L_j, $1 \leq j \leq L_d$. The prediction then means that the tool predicts whether or not a failure will occur in the next ten days ($j + 10$ days).

$$P_{i,j} \in \{0,1\} \tag{6.16}$$

$$P_{i,j} = \begin{cases} 1 & \text{if tool predicts on day } j \text{ that the component of AC } i \text{ will break in the next } L_d - j \text{ days} \\ 0 & \text{if tool predicts on day } j \text{ that the component of AC } i \text{ will not break in the next } L_d - j \text{ days} \end{cases} \tag{6.17}$$

At day 0 the values for $P_{i,j}$ are generated in advance and the values between day 1 and day L_d are stored in the respective layers L_j. Whereas the predictions of the days prior to day 0 are stored in a separate matrix P_p, where each row presents a layer L_j, $L_p \leq j \leq -1$, and each column shows the predictions of an aircraft i, $i \in I$.

$$P_p = \begin{bmatrix} P_{1,-1} & P_{2,-1} & ... & P_{I,-1} \\ p_{1,-2} & P_{2,-2} & ... & P_{I,-2} \\ P_{1,-3} & P_{2,-3} & ... & P_{I,-3} \\ P_{1,-4} & P_{2,-4} & ... & P_{I,-4} \\ ... & ... & ... & .... \\ P_{1,L_p} & P_{2,L_p} & ... & P_{I,L_p} \end{bmatrix} \tag{6.18}$$

The bias used to randomly create the predictions is linked to the failure function in Figure 5.2. Meaning that a component with a long time since last repair t_r is more likely to present positive failure predictions.

6.1.7. Counter

The counter x_{j,k_j} indicates how often a specific node n_{j,k_j} with k_j, $1 \le k_j \le n^j_{child}$ on layer L_j, $1 \le j \le L_d$ has been included in an iteration.

$$x_{j,k_j} = \begin{cases} 0 & \text{at initialization} \\ 1 & \text{after first random simulation} \\ 1 + \sum\limits_{u=n_{child}\cdot(k-1)+1}^{k\cdot n_{child}} x_{j+1,u} & \text{if } x_{j,k} \ge 1 \end{cases} \tag{6.19}$$

Each counter is, therefore, a summation of the first iteration and the counters of the child nodes. Using this method as can be seen in Figure 6.8 all related counter values x_{j,k_j} are automatically updated.

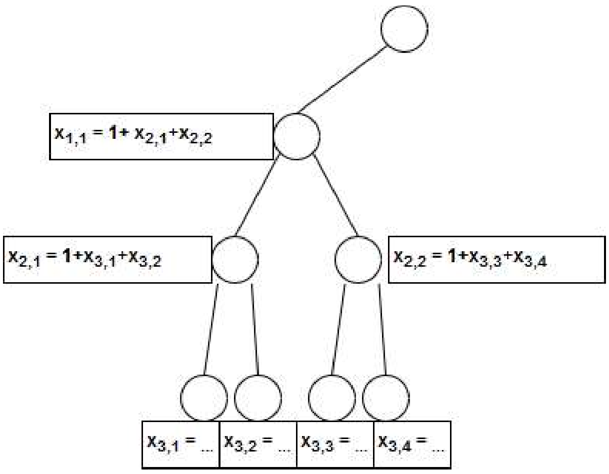


Figure 6.8: Sample counter computations

The root node n_0 is also equipped with a counter x_0 which adheres to similar concepts as the counter on other layers x_{j,k_j}, meaning that it is either 0 at initialization or the sum of all counters of its child nodes.

$$x_0 = \begin{cases} 0 & \text{at initialization} \\ \sum\limits_{u=n_{child}\cdot(k-1)+1}^{k\cdot n_{child}} x_{j+1,u} & \text{after expansion} \end{cases} \tag{6.20}$$

6.1.8. Infeasibility Flag

As stated above there are many cases where a child node is not a feasible solution to the problem. To cope with the numbering scheme of the actions and nodes, an infeasibility flag is incorporated to know when the node does not require further simulation. A flag is defined for each node n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$ and $1 \leq j \leq L_d$.

$$F_{j,k_j} \in \{0,1\} \tag{6.21}$$

$$F_{j,k_j} = \begin{cases} 0 & \text{if node } n_{j,k_j} \text{ is not a feasible node} \\ 1 & \text{if node } n_{j,k_j} \text{ is a feasible node} \end{cases} \tag{6.22}$$

It is dependent on the actions a_{i,j,k_j}, the states $S_{i,j-1,\left\lceil \frac{k}{n_{child}} \right\rceil}$ of the parent node $n_{j-1,k_{j-1}}$ and the maintenance opportunities $m_{i,j}$.

$$F_{j,k_j}\left(a_{i,j,k_j}, S_{i,j-1,\left\lceil \frac{k}{n_{child}} \right\rceil}, m_{i,j}\right) = \begin{cases} 0 & \text{if } a_{i,j,k_j} = 1 \textbf{ and } S_{i,j-1,\left\lceil \frac{k}{n_{child}} \right\rceil} = 1 \quad \text{for any } i \in I \\ 0 & \text{if } a_{i,j,k_j} = 1 \textbf{ and } m_{i,j} = 0 \quad \text{for any } i \in I \\ 0 & \text{if } \sum_{i=1}^{I} a_{i,j,k_j} > M_j \quad \text{for any } i \in I \\ 0 & \text{if } j = L_d \\ 1 & \text{all other cases} \end{cases} \tag{6.23}$$

6.2. Simulation

Each node n_{j,k_j} is evaluated based on the potential shown during random rollout simulations. During those simulations, a random combination of actions and classification component prognostics are simulated.

6.2.1. Simulated Action

Unlike the actual action a_{i,j,k_j}, the simulated action $\tilde{a}_{i,\tilde{j}}$ is not a node property but an action which is randomly chosen in the simulation process for each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$ and for each aircraft i, $i \in I$.

Note: The index j is now a set input constant (layer L_j of the start of the simulation) for a simulation and unlike before a changeable variable.

$$\tilde{a}_{i,\tilde{j}} \in \{0,1\} \tag{6.24}$$

$$\tilde{a}_{i,\tilde{j}} = \begin{cases} 1 & \text{if the random simulation chooses to repair AC } i \text{ on day } \tilde{j} \\ 0 & \text{if the random simulation chooses not to repair AC } i \text{ on day } \tilde{j} \end{cases} \tag{6.25}$$

Which action $\tilde{a}_{i,\tilde{j}}$ is chosen in the simulation for aircraft i on layer $\tilde{j}$ is constrained by similar aspects as the actual actions a_{i,j,k_j} and therefore it is dependent on the state of the leaf node S_{i,j,k_j} and the maintenance opportunities of the simulated layers $m_{i,\tilde{j}}$.

Note: The index k_j is now a set input constant (node n_{j,k_j} where the random simulation starts) for a simulation and unlike before a changeable variable.

$$\tilde{a}_{i,\tilde{j}}\left(S_{i,j,k_j}, m_{i,\tilde{j}}\right) = \begin{cases} 0 & \text{if } S_{i,j,k_j} = 1 \\ 0 & \text{if } m_{i,\tilde{j}} = 0 \\ 0 \text{ or } 1 & \text{if } S_{i,j,k_j} = 0 \textbf{ and } m_{i,\tilde{j}} = 1 \end{cases} \tag{6.26}$$

And *all other cases* are therefore constrained by a maximum amount of possible simultaneous repairs on layer L_j (M_j) and the fact that each aircraft should only be repaired once in L_d days.
It is defined for each layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, that at most only M_j aircraft can be repaired:

$$\sum_{i=1}^{I} \tilde{a}_{i,\tilde{j}} \leq M_j \tag{6.27}$$

And it is defined for each aircraft i, $i \in I$, that, it can be repaired at most only one time between day j and day L_d:

$$\sum_{\tilde{j}=j}^{L_d} \tilde{a}_{i,\tilde{j}} \leq 1 \tag{6.28}$$

In the case of $S_{i,j,k_j} = 0$ **and** $m_{i,\tilde{j}} = 1$ as stated in equation 6.26 a biased random generator simulates a 1 or 0 decision:

$$\tilde{a}_{i,\tilde{j}} = \begin{cases} Prob(\tilde{a}_{i,\tilde{j}} = 1) = 0.31 & \text{if } P_{i,j} = 1 \\ Prob(\tilde{a}_{i,\tilde{j}} = 0) = 0.86 & \text{if } P_{i,j} = 0 \end{cases} \tag{6.29}$$

6.2.2. Random Simulation

At day j for all aircraft i for all simulated days $\tilde{j}$, $j \leq \tilde{j} \leq L_d$ a random selection of simulated actions $\tilde{a}_{i,\tilde{j}}$ based on equations 6.26, 6.27 and 6.28 is generated. How well the simulated action compares with the prediction is computed in the rollout values and the punishments as explained below.

6.2.3. Award and Rollout Value Computations

Each node n_{j,k_j} is furthermore equipped with an award value A_{j,k_j} and a random rollout value R_{j,k_j}. The random rollout value is computed using a random simulation of the next $L_d - j$ days and it is restricted by a maximum possible rollout value R_m.

$$A_{j,k_j} \in \mathbb{R}^+ \tag{6.30}$$

$$R_{j,k_j} \in \mathbb{R}^+ \text{ , } (0 \leq R_{j,k_j} \leq R_m) \tag{6.31}$$

$$A_{j,k_j} = \begin{cases} 0 & \text{if } x_{j,k_j} = 0 \\ R_{j,k_j} & \text{if } x_{j,k_j} = 1 \\ R_{j,k_j} + \sum\limits_{u=n_{child}\cdot(k-1)+1}^{k\cdot n_{child}} A_{j+1,u} & \text{if } x_{j,k_j} > 1 \end{cases} \tag{6.32}$$

As can be seen in equation 6.32, when a node is initiated the award value A_{j,k_j} is set to zero. Before a simulated rollout is performed both the counter x_{j,k_j} and the award value A_{j,k_j} are still equal to zero. Once the simulated rollout is performed the counter x_{j,k_j} updates to 1 and the award value A_{j,k_j} to the newly obtained rollout value R_{j,k_j}. After which the award value A_{j,k_j} will be updated after each iteration by adding the newly obtained rollout values along that branch of the tree, by summing the award values of the n_{j,k_j}'s child nodes.

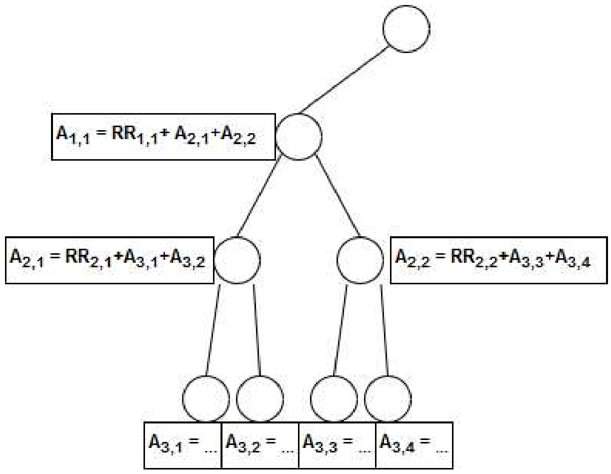


Figure 6.9: Sample Award Value Computations

The root node n_0 is also equipped with an award value A_0, which is based on similar concepts as the previously defined award values on other layers A_{j,k_j}. Therefore it is also equal to zero when the root node has not been part of an iteration yet $x_0 = 0$ and it is the sum of the award values of its child nodes (see equation 6.34).

$$A_0 \in \mathbb{R}^+ \tag{6.33}$$

$$A_0 = \begin{cases} 0 & \text{if } x_0 = 0 \\ \sum_{u=1}^{n_{child}} A_{j+1,u} & \text{if } x_0 \geq 1 \end{cases} \tag{6.34}$$

The random rollout simulation uses a random combination of yes and no predictions and simulated yes and no repair actions to compute the potential of choosing a specific node.

The random rollout value can be computed using a maximum possible rollout value R_m subtracted by the sum of all punishments n, $1 \leq n \leq N$, on node n_{j,k_j}, with $1 \leq k_j \leq n_{child}$ and $1 \leq j \leq L_d$, per aircraft i, $i \in I$.

The maximum possible random rollout value R_m is dependent on the amount of punishments N, number of days into the past L_d and the amount of aircraft I.

$$R_m = N \cdot (L_d - 1) \cdot I \tag{6.35}$$

At each n_{j,k_j}, R_{j,k_j} is defined as:

$$R_{j,k_j} = \begin{cases} 0 & \text{if } x_0 = 0 \\ R_m - \sum_{n=1}^{N} \sum_{\tilde{j}=j}^{L_d} \sum_{i=1}^{I} C_{n,\tilde{j},i} & \text{if } x_0 \geq 1 \end{cases} \tag{6.36}$$

Therefore a rollout value R_{j,k_j} close to R_m shows a node with positive potential and few punishments in the random simulation.

6.3. Punishment Structure

6.3.1. Punishment Overview

The punishment values $C_{n,\tilde{j},i}$ are defined on each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, for each punishment n, $1 \leq n \leq N$, and for each aircraft i, $i \in I$.

$$C_{n,\tilde{j},i} \in \mathbb{R}^+ \text{ , } (0 \leq C_{n,\tilde{j},i} \leq 1) \tag{6.37}$$

Punishments are the basis for the random rollout values R_{j,k_j} and therefore also of the award values A_{j,k_j}. The initial model will include three punishments ($N = 3$) but they will be extended in future updated model versions. The numerical value of the punishments is based on the accuracy of the used prognostics tool as explained at the beginning of this chapter.

6.3.2. Average Lifetime Punishments

This punishment consists of two parts namely firstly punishing a repair action if the time since last repair t_{r_i} is shorter than the average replacement interval $t_{\bar{a}}$ and secondly punishing no repair within L_d days even though the time since last repair t_{r_i} is larger than the average replacement interval $t_{\bar{a}}$. The average replacement interval is computed from data related to the time between unscheduled maintenance actions.

Punishment 1 $C_{1,\tilde{j},i}$ is defined on each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, and for each aircraft i, $i \in I$, with the requirement being that the time since last repair t_{r_i} is shorter than the average replacement interval $t_{\bar{a}}$:

$$C_{1,\tilde{j},i} = \begin{cases} 1 - \dfrac{t_r}{t_{\bar{a}}} & \text{if } t_r < t_{\bar{a}} \text{ and } S_{i,j,k_j} = 1 \\ 1 - \dfrac{t_r}{t_{\bar{a}}} & \text{if } t_r < t_{\bar{a}} \text{ and } a_{i,\tilde{j}} = 1 \\ 0 & \text{if } t_r \geq t_{\bar{a}} \\ 0 & \text{if } S_{i,j,k_j} = 0 \text{ and } a_{i,\tilde{j}} = 0 \end{cases} \tag{6.38}$$

Punishment 2, $C_{2,j,i}$, is defined for the sequence of L_p days, for each aircraft i, $i \in I$, with the requirement being that the time since last repair t_{r_i} is larger than the average replacement interval $t_{\bar{a}}$. The punishment is allocated to day j.

$$C_{2,\tilde{j},i} = \begin{cases} 1 - \dfrac{t_{\bar{a}}}{t_r} & \text{if } t_r > t_{\bar{a}} \text{ and } S_{i,j,k_j} = 0 \text{ and } a_{i,\tilde{j}} = 0 \\ 0 & \text{if } t_r \leq t_{\bar{a}} \\ 0 & \text{if } S_{i,j,k_j} = 1 \\ 0 & \text{if } a_{i,\tilde{j}} = 1 \end{cases} \tag{6.39}$$

6.3.3. Simulated Prognostics Punishment

This punishment relates the simulated predictions $P_{i,j}$ at a simulated day $\tilde{j}$ (between day 1 and day L_d) with the simulated action $\tilde{a}_{i,\tilde{j},k_j}$ at the same day. Meaning that a simulated action is punished if it does not match the simulated prediction on a specific day. The punishment is based on the accuracy of yes or no predictions of the used prognostics tool and therefore the punishment is higher is the accuracy is higher.

The second punishment value $C_{3,\tilde{j},i}$ is defined on each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, and for each aircraft i, $i \in I$.

$$C_{3,\tilde{j},i}(P_{i,j},\tilde{a}_{i,\tilde{j}}) = \begin{cases} NPV & \text{if } P_{i,j}=0 \text{ and } \tilde{a}_{i,\tilde{j}}=1 \\ (1-NPV) & \text{if } P_{i,j}=0 \text{ and } \tilde{a}_{i,\tilde{j}}=0 \quad \text{, if } NPV \geq 0.5 \\ 0 & \text{if } P_{i,j}=0 \text{ and } \tilde{a}_{i,\tilde{j}}=0 \quad \text{, if } NPV < 0.5 \\ (1-PPV) & \text{if } P_{i,j}=1 \text{ and } \tilde{a}_{i,\tilde{j}}=1 \quad \text{, if } PPV \geq 0.5 \\ 0 & \text{if } P_{i,j}=1 \text{ and } \tilde{a}_{i,\tilde{j}}=1 \quad \text{, if } PPV < 0.5 \\ PPV & \text{if } P_{i,j}=1 \text{ and } \tilde{a}_{i,\tilde{j}}=0 \end{cases} \tag{6.40}$$

6.3.4. Previous Prognostics Punishment

This punishment includes the 'known' predictions made prior to initialization of the model. It punishes a sequence of 10 days if the amount of repairs in those 10 days does not match the prediction made 10 days ago. The punishment is awarded on the last day of that 10-day sequence. The punishment is also related to the accuracy of the prognostics model, meaning that the punishment is higher if the accuracy is higher.
The third punishment $C_{4,\tilde{j},i}$ is defined on each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$ depending on the actions $\tilde{a}_{i,j}$ of the previous 10 days $\tilde{j}-10,....,\tilde{j}$, and for each aircraft i, $i \in I$.

$$C_{4,\tilde{j},i}(\tilde{a}_i,\tilde{j},P_{i,\tilde{j}}) = \begin{cases} PPV & \text{if } P_{i,j-10}=1 \text{ and } S_{i,j,k_j}=0 \text{ and } a_{i,\tilde{j}}=0 \\ (1-PPV) & \text{if } P_{i,j-10}=1 \text{ and } (S_{i,j,k_j}=1 \text{ or } a_{i,\tilde{j}}=1) \text{ and } PPV \geq 0.5 \\ 0 & \text{if } P_{i,j-10}=1 \text{ and } (S_{i,j,k_j}=1 \text{ or } a_{i,\tilde{j}}=1) \text{ and } PPV < 0.5 \\ NPV & \text{if } P_{i,j-10}=0 \text{ and } (S_{i,j,k_j}=1 \text{ or } a_{i,\tilde{j}}=1 \\ (1-NPV) & \text{if } P_{i,j-10}=0 \text{ and } S_{i,j,k_j}=0 \text{ and } a_{i,\tilde{j}}=0 \text{ and } NPV \geq 0.5 \\ 0 & \text{if } P_{i,j-10}=0 \text{ and } S_{i,j,k_j}=0 \text{ and } a_{i,\tilde{j}}=0 \text{ and } NPV < 0.5 \end{cases} \tag{6.41}$$

6.3.5. Expensive Maintenance Slot Punishment

This punishment is related to maintenance opportunities. It punishes a simulated maintenance action if the action is planned on a day with a more expensive maintenance slot. An airline maintenance schedule differentiates between specifically allocated maintenance slots of a specific aircraft and generally available maintenance opportunities for a type of aircraft. Since the aircraft is already taken of the active operational schedule for the fixed maintenance slots and the fact that maintenance technicians are already scheduled it is cheaper to perform a repair then compared to a generally available slot.

The total amount of maintenance opportunities TM_i and the amount of specific maintenance slots SM_i are defined for aircraft i, $i \in I$ between day 0 and day L_d:
Let SM_i be the set of all specific maintenance actions of $m_{i,1},...,m_{i,L_d}$ with $m_{i,j}=2$. And let TM_i be the set of all maintenance opportunities of $m_{i,1},...,m_{i,L_d}$ with $m_{i,j}>0$.

$$SM_i = \sum_{j=1}^{L_d} \mathbf{1}_{m_{i,\tilde{j}}=2} \tag{6.42}$$

$$TM_i = \sum_{j=1}^{L_d} \mathbf{1}_{m_{i,\tilde{j}}>0} \tag{6.43}$$

The fifth punishment $C_{5,\tilde{j},i}$ is defined on each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, and for each aircraft i, $i \in I$.

$$C_{5,\tilde{j},i}(m_{i,\tilde{j}}, \tilde{a}_i, \tilde{j}) = \begin{cases} \frac{SM_i}{TM_i} & \text{if } \tilde{a}_{i,\tilde{j}} = 1 \text{ and } m_{i,\tilde{j}} = 1 \\ 0 & \text{if } \tilde{a}_{i,\tilde{j}} = 0 \\ 0 & \text{if } m_{i,\tilde{j}} = 2 \\ 0 & \text{if } m_{i,\tilde{j}} = 0 \end{cases} \tag{6.44}$$

6.4. MCTS Iteration Steps

The MCTS consists of four steps which are explained below.

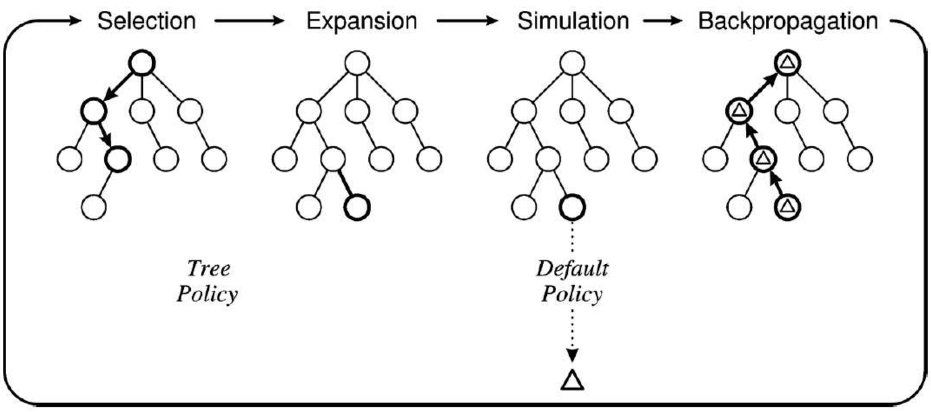


Figure 6.10: Steps of one iteration of the MCTS approach

6.4.1. Selection

The Monte-Carlo tree search tries to balance its iterations between exploration and exploitation to cover the nodes with the highest potential with the least computational effort. Meaning that it selects nodes n_{j,k_j} with high award values A_{j,k_j} and with few visits x_{j,k_j}. The Upper Confidence Bounds (UCB1) algorithm can be used to select a node. A trade off is made between exploitation (*Term:* $\overline{A}_{j,k}$) and exploration (*Term:* $\sqrt{\frac{ln(x_0)}{x_{j,k}}}$, where x_0 is the counter of the root node). A weighting constant c_w is used to balance those aspects and can be empirically tuned depending on whether one prefers exploitation or exploration, in literature it is usually a positive single digit value ($0 < c_w < 10$).
The UCB1 value $UCB1_{j,k_j}$ is therefore defined for each node n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$ and $1 \leq j \leq L_d$:

$$UCB1_{j,k_j} = \overline{A}_{j,k_j} + c_w \sqrt{\frac{ln(x_0)}{x_{j,k_j}}} \tag{6.45}$$

The average award value of node n_{j,k_j}, with $1 \leq k_j \leq n^j_{child}$ and $1 \leq j \leq L_d$, can be calculated dividing the award value A_{j,k_j} by the amount of visits of node n_{j,k_j}.

$$\overline{A}_{j,k_j} = \frac{A_{j,k_j}}{x_{j,k_j}} \tag{6.46}$$

The UCB1 value needs to be calculated for all child nodes in question such that the child node with the highest UCB1 value can be selected as the most promising one for the next simulation iteration as shown in equation 6.47.

$$\text{select } n_{j,k_j} \text{ with } max(UCB1_{j,k_j}) \tag{6.47}$$

To summarize:

1. Compute $UCB1_{j,k_j}$ value for all nodes
   (a) Calculate average award value $\overline{A}_{j,k_j}$ up until the node of interest
   (b) Determine counter of the root node x_0 and counter of the node of interest x_{j,k_j}
2. Select the child node n_{j,k_j} with the highest UCB1 value $UCB1_{j,k_j}$

6.4.2. Expansion

If the selected node has been visited once before, $x_{j,k_j} = 1$, it needs to be expanded. This means that all n_{child} possible child nodes are added into the tree and they get initialized with all node statistics as explained before.
In case the node has not been visited before, $x_{j,k_j} = 0$, this step is skipped and the node is directly rolled out in a random simulation as explained in the next step.

3. $\begin{cases} \text{if } x_{j,k_j} = 1 & \text{see step 4 and expand the node } n_{j,k_j} \\ \text{if } x_{j,k_j} = 0 & \text{see step 5 and perform random simulation of node } n_{j,k_j} \end{cases}$
4. Expand the node by initializing all n_{child} possible child nodes (each child node equipped with all node information)

6.4.3. Simulation

Simulation can start at a feasible node n_{j,k_j} which has not been rolled out before ($F_{j,k_j} = 1$ and $x_{j,k_j} = 0$). When such a node is found it can be rolled out in a random simulation which means that for each simulated layer $\tilde{j}$, $j \leq \tilde{j} \leq L_d$, and for each aircraft i, $i \in I$, a random simulated action $\tilde{a}_{i,\tilde{j}}$ is compared with the prediction $P_{i,j}$ made for that layer. The comparison is determined and documented in the punishments $C_{n,\tilde{j},i}$, which will then be used in the rollout value R_{j,k_j} and award value A_{j,k_j} as defined in equations 6.36 and 6.32 respectively.

5. Confirm that simulation can start on the selected node ($F_{j,k_j} = 1$ and $x_{j,k_j} = 0$)
6. Create a random combination of simulated actions $\tilde{a}_{i,\tilde{j}}$ for the next $L_d - j$ days according to the concept defined in equations 6.26, 6.27, 6.28.
7. Determine all punishments $C_{n,\tilde{j},i}$ for the given combination of predictions $P_{i,j}$ and simulated actions $\tilde{a}_{i,\tilde{j}}$ as stated in equations 6.38, 6.40, 6.41.
8. Compute the rollout value R_{j,k_j} according to equation 6.36.
9. Compute the award value A_{j,k_j} according to equation 6.32.

6.4.4. Backpropagation

Once a rollout value R_{j,k_j} is found for node n_{j,k_j} the award value A_{j,k_j} and the counter x_{j,k_j} of that node are updated according to the concept explained in equation 6.32 and 6.19. This automatically results in updating the award values and counters along that branch according to the concept explained previously (see Figures 6.7 and 6.8).

10. Update the rollout value R_{j,k_j} of the previously rolled out node n_{j,k_j}
11. Update the award value A_{j,k_j} of the previously rolled out node n_{j,k_j}

12. Update the counter x_{j,k_j} of the previously rolled out node n_{j,k_j}
13. All other award values and counters up until this node are automatically updated according to the concept defined in equations 6.32 and 6.19
14. Start new iteration at step 1

7
Additional Results

The results and findings of the main model and two interesting changes to the model were previously discussed in Part I of the report. These results of the Monte-Carlo tree search maintenance scheduling model presented in Part I showed interesting and seemingly robust and correct results. It is furthermore also interesting to analyze the possibility to schedule aircraft of different types simultaneously, since most airlines do not only operate just one type of aircraft but multiple. If one allows the different types to plan repair actions also during slots of the other type it is possible to schedule multiple aircraft of the same type at the same day. Due to a large increase in computational runtime it is not possible to present a schedule for the full set of 20 aircraft as done before. This analysis focuses on four aircraft of type 1 and four aircraft of type 2. Table 7.1 presents the maintenance schedule when the extension of having two types is implemented. Whereas the result of individually scheduling the first four aircraft with just their own available maintenance slots and then the other four aircraft with their general available type maintenance slots can be seen in Table 7.2.

Table 7.1: Scheduled Repair Actions 8 Aircraft of different types (scheduled together)

AC 1			AC 2			AC 3			AC 4		
type 1			type 1			type 1			type 1		
t_r = 178			t_r = 245			t_r = 233			t_r = 657		
m_ij		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*
Day	*Type*		*Day*	*Type*		*Day*	*Type*		*Day*	*Type*	
2	GM	✓	2	OGM	-	3	GM	-	1	GM	-
3	GM	-	3	GM	✓	6	OGM	-	2	OGM	✓
5	GM	-	5	OGM	-	9	OGM	✓	5	GM	-
6	OGM	-	6	OGM	-				7	GM	
7	OGM	-	9	OGM	-						

AC 5			AC 6			AC 7			AC 8		
type 2			type 2			type 2			type 2		
t_r = 536			t_r = 205			t_r = 430			t_r = 543		
m_ij		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*
Day	*Type*		*Day*	*Type*		*Day*	*Type*		*Day*	*Type*	
1	OGM	✓	1	OGM	-	3	GM	-	1	OGM	-
3	OGM	-	3	GM	✓	6	GM	✓	2	GM	-
7	OGM	-	6	GM	-	9	GM	-	5	OGM	✓
10	GM	-	7	GM	-				7	OGM	-
			9	GM	-				10	GM	-
			10	GM	-						

Table 7.2: Scheduled Repair Actions 8 Aircraft of different types (scheduled individually)

AC 1			AC 2			AC 3			AC 4		
type 1			type 1			type 1			type 1		
t_r = 178			t_r = 245			t_r = 233			t_r = 657		
m_ij		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*
Day	*Type*		*Day*	*Type*		*Day*	*Type*		*Day*	*Type*	
2	GM	✓	3	GM	-	3	GM	-	1	GM	✓
3	GM	-							5	GM	-
5	GM	-							7	GM	-

AC 5			AC 6			AC 7			AC 8		
type 2			type 2			type 2			type 2		
t_r = 536			t_r = 205			t_r = 430			t_r = 543		
m_ij		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*	*m_ij*		*Repair?*
Day	*Type*		*Day*	*Type*		*Day*	*Type*		*Day*	*Type*	
10	GM	-	3	GM	-	3	GM	✓	2	GM	✓
			6	GM	✓	6	GM	-	10	GM	-
			7	GM	-	9	GM	-			
			9	GM	-						
			10	GM	-						

We can see that Table 7.1 shows that all aircraft have more maintenance slots available to them and that indeed multiple aircraft are scheduled during an opportunity of the other type.

Unfortunately by extending the maintenance scheduling mode with 2 types of aircraft, the feasible search space of the Monte-Carlo tree search is increased drastically. Which furthermore results in long computational runtimes which makes it in the current form less applicable in an operational airline context.
The extension with 2 types in its' current form furthermore seems to schedule a lot of aircraft because there are the opportunities to do so even though they were not scheduled when not presenting them with the option of using alternative maintenance slots. It seems as the time since last repair is taken into account less then when there is a limited maintenance slot availability.

8
Discussion

8.1. Main Maintenance Scheduling Model

Scheduling Results

The main objective of the model is to schedule multiple aircraft simultaneously for a pre-defined amount of days into the future. This needs to be done using classification prognostics and available maintenance slots into account. As shown in Part I this is indeed the results of the proposed model, which is also done is a comparatively accurate manner. Most of the aircraft scheduled for maintenance do have a time since last repair above the mean time between repairs. There are however also some aircraft which have a shorter time since last repair but are scheduled either way and others with a high time since last repair which are not scheduled.

These differences between the scheduling results and what would be expected can be explained by the way the Monte-Carlo tree search makes its decision in the selection phase of each iteration. The selection criterion selects a node with high awards and/or few visits during iterations. The current model does not include a check whether this is actually is the best node. Especially in the final iteration this check is missing in order to guarantee that the most logical aircraft are scheduled for repair.

It can however nicely be seen that the model indeed favours specific maintenance slots over general maintenance slots when scheduling maintenance. This nicely shows that the distinction between cheaper and more expensive maintenance slots improves the maintenance scheduling model towards a cheaper overall solution.

Robustness

A positive result of the main model is also the high robustness of the main maintenance scheduling model. Presenting the model with the same input of time since last repairs, available maintenance slots and prognostics of previous days leads to a similar result. Meaning that almost identical aircraft are scheduled and that the schedules only sometimes deviate on which days the aircraft is scheduled for maintenance. In the presented case that 2 aircraft switch on which days they are scheduled has the effect that the airline do know on which days the maintenance actions is scheduled but not for which aircraft. They are however scheduled on specific maintenance slots which means that it is already scheduled out of operations and thus no negative effect for the airline can be expected from this slight uncertainty. The fact that one aircraft is not consistently scheduled during all runs has however more negative effects on an airline. These are however last drastically because the maintenance is scheduled at the end of the scheduling horizon and therefore new prognostics insight would improve the decision making. This overall good performance with respect to robustness allows the therefore conclusion that the model will also be able to robustly schedule larger amount of aircraft.

Runtime

The up-scaling of the case study with larger fleet sizes is however hindered by the steeply increasing computational runtimes for larger amount of aircraft. The reason for this is the large increase of possible child nodes and therefore the possible options to evaluate. Up until around 20 AC of the same type the runtime is, however, feasible for an operational context. Most airlines operate varying amount of aircraft of the same type and therefore many of those cases are below 20 aircraft but others are also above and therefore making the current model implementation less useful for those situations.

8.2. Probability Failure Distribution Prognostics

Expanding the model with failure distribution as a prognostics input instead of component prognostics as presented in Part I nicely showed that the chosen model is also able to incorporate probabilities in the decision making process. Adding failure distribution probabilities add additional knowledge to the maintenance scheduling model. The current version implements an approximated failure distribution and therefore no accuracy information of those probabilities are included.

8.3. No Simulated Prognostics

Removing the simulated prognostics and just considering the known prognostics of the previous days and day 1 shows that the results are almost identical as before. This means that the simulated prognostics and the related punishment to that do not have a large effect on the maintenance scheduling process.

8.4. Two Types of Aircraft

Changing the model in order to be able to incorporate multiple types of aircraft while allowing them to be scheduled on maintenance slots of the other type presented an interesting case study but proved to be not yet sufficiently developed. The current runtimes only allow for extremely small fleet sizes to be scheduled in a reasonable time frame. The results furthermore showed that the current model schedules maintenance progressively, meaning that it schedules also aircraft with low time since last repairs. This behaviour is not yet sufficient for an airline application.

It is however also clear that the model extension is only required if one wishes to allow aircraft to select maintenance slots of other types. If one would simply schedule multiple types of aircraft with their own maintenance slots no changes to the model would be required and the original main model could be applied.

9 Verification and Validation

In order to ensure the correctness of the Monte-Carlo tree search maintenance scheduling model, it is important to verify and validate the model and its' results. Verification as described in Section 9.1 analyses whether the maintenance scheduling model performs the computations correctly. Then the validation in Section 9.2 evaluates whether the model appropriately represents the actual operational context.

9.1. Verification

The verification of the maintenance scheduling model is performed in a unit and a system analysis in order to ensure that all parts of the model and also the full model perform as expected.

During the one-step analysis, each part of the model and its implementation are walked through to confirm that the step is indeed defined correctly. This is done using careful logical analysis and testing each step with simplified and standard input parameters (no maintenance slots, always maintenance slot, no failure predictions, always yes failure predictions, extreme accuracy values). Whenever a step not presents the expected output the step is reviewed and improved followed by another analysis until the final model at its' implementation behaves as expected when confronted with extreme input.

A similar process is used to verify the complete model. Analyzing each connection between individual steps and their order in the overall model guarantees that not just the specific units but also the overall system performs as expected. This is again confirmed with an extreme value analysis of the whole model.

9.2. Validation

Validation of the maintenance scheduling model is performed in the form of a case study with airline data combined with artificially generated data.

Using the input data as presented in Part I the maintenance scheduling is performed and the results are analyzed. By evaluating that indeed no repairs are scheduled on days without maintenance slots and that aircraft with appropriate time since last repair times are scheduled shows that the maintenance scheduling model is indeed able to correctly represent the actual airline maintenance operation.

Unfortunately, no maintenance scheduling models using prognostics and available maintenance slots of multiple aircraft are currently applied at airlines, therefore, it is not possible to compare the results of the proposed maintenance scheduling model with existing and therefore verified and validated models.

10 Conclusions and Recommendations

As stated before, the Monte-Carlo tree search maintenance scheduling algorithm is able to robustly schedule multiple aircraft based on their failure prognostics. The main issue of the current model and its' extension is however the run time. As stated in the literature review, many other scheduling models only start the planning process once an aircraft is triggered due to high time since last repair or low remaining useful life. The deliberate choice of considering all aircraft and letting the model compute which ones to repair ensures that all possible options are considered but is also the main reason for the high computational run times. The main recommendation is therefore to work on reducing the computational time for larger sets of aircraft. One might look into implementing some form of trigger to remove aircraft not fulfilling any of the criteria required in order to be scheduled or otherwise increase searching efficiency. But it is also interesting to evaluate different selection procedures in the MCTS steps or to improve the current UCB1 procedure. The current implementation uses the default value of $\sqrt{2}$ as a factor between exploitation and exploration but other tuned factors might improve search efficiency while delivering similar maintenance schedules.
The fact that in the final iteration an aircraft with a clearly lower time since last repair was scheduled instead of others also showed that the selection procedure is not yet perfect. In the final iteration a check should be implemented to ensure that not a node with an award value comparatively too low is selected only because of its' few visits during iterations.

The failure distribution probability case study offers an interesting solution for further developments. Probability prognostics create a more complete set of information about the health status of a component which makes it more useful for further implementations. It is however required to further improve the model in order to include the accuracy of the failure predictions which is not implemented in the maintenance scheduling model.
It will also be interesting to compare the effect between continuous probability prognostics (increase in failure probability per day) and just one failure probability. The first also offers the possibility to apply a clear objective function and optimization of the maintenance scheduling whereas the latter will most likely be less suited in the form of a Monte-Carlo tree search implementation since there is no trend of prognostics which will largely increase the search space and therefore the runtime will increase further.
The analysis of neglecting simulated prognostics showed that this is a feasible solution which presents almost identical maintenance schedules as the main model without assuming prognostics input. Further improvements of the MCTS maintenance scheduling model should therefore use this as a standard setup.
Implementing the multiple type extension is a nice proof of concept but is currently not yet done this way in the airline industry. Currently airlines only use the maintenance slots of the actual aircraft type instead of using those of other types. This model showed that it might be interesting topic to look into from an operational and from a scheduling point of view. If one wishes to further implement this concept is is definitely required to improve the computational efficiency first and to

analyse and improve the behaviour of the model if it is offered with many available maintenance slots.

One possibility provided by the chosen Monte-Carlo tree search approach is the concept of a moving horizon. This is currently not yet implemented but would be interesting to analyse since it results in a more realistic solution. The current model once schedules all 20 aircraft for the upcoming ten days but a moving horizon scheduling approach would be able to make use of an additional knowledge of days once they have passed. By scheduling each day for the upcoming ten days while implementing the new data of the last day would improve the usability and accuracy of the model further.
Another aspect which should be analysed if one wishes to improve real life airline operations applicability, is the fact that the current maintenance scheduling model assumes each maintenance slot be a full day and that is is always available to schedule an additional maintenance action, which is clearly a strong assumption.
A Monte-Carlo tree search approach was chosen due to the simplified prognostics input available and it proved to a suitable method for a simplified input environment. For future developments one should aim however at obtaining more sophisticated prognostics data as an input to the planning framework. Linking actual and more accurate prognostics information and the maintenance availability of the respective aircraft one would be able to develop a more accurate model. This would then also make the implementation at an airline easier.

Bibliography

[1] P. Sparaco. LCCs on route to dominate european point-to-point travel. *Aviation Week and Space Technology (New York)*, 173(19):46, 2018.

[2] X. Chen, S. Wang, B. Qiao, and Q. Chen. Basic research on machinery fault diagnostics: Past, present, and future trends. *Frontiers of Mechanical Engineering*, 13(2):264–291, 2018.

[3] T. Sutharssan, S. Stoyanov, C. Baley, and C. Yin. Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. *The Journal of Engineering*, 2015(7):215–222, 2015.

[4] L. Liao and F. Köttig. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1):191–207, 2014.

[5] J. Luo, M. Namburu, K. Pattipati, L. Qiao, M. Kawamoto, and S. Chigusa. Model-based prognostic techniques. In *IEEE Systems Readiness Technology Conference*, pages 330–340, 2003.

[6] H. Sun, D. Cao, Z. Zhao, and X. Kang. A hybrid approach to cutting tool remaining useful life prediction based on the wiener process. *IEEE Transactions on Reliability*, 67(3):1294–1303, 2018.

[7] T. Biagetti and E. Sciubba. Automatic diagnostics and prognostics of energy conversion processes via knowledge-based systems. *Energy*, 29(12-15):2553–2572, 2004.

[8] A. Majidian and M.H. Saidi. Comparison of fuzzy logic and neural network in life prediction of boiler tubes. *International Journal of Fatigue*, 29(3):489–498, 2007.

[9] Y. Li, S. Billington, C. Zhang, T. Jurfess, S. Danyluk, and S. Liang. Adaptive prognostics for rolling element bearing condition. *Mechanical Systems and Signal Processing*, 14(1):103–113, 1999.

[10] K. Sobczyk and B. Spencer. Random fatigue: From data to theory. *Journal of Engineering Mechanics*, 119(2):415–416, 1993.

[11] S. Yang and T. Liu. State estimation for predictive maintenance using kalman filter. *Reliability Engineering & System Safety*, 66(1):29–39, 1999.

[12] S. Yang. An experiment of state estimation for predictive maintenance using kalman filter on a dc motor. *Reliability Engineering & System Safety*, 75(1):103–111, 2002.

[13] K. Sobczyk and J. Trebicki. Stochastic dynamics with fatigue-induced stiffness degradation. *Probabilistic Engineering Mechanics*, 15(1):91–99, 2000.

[14] A. Ray and S. Tangirala. Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, 4(4):413–422, 1996.

[15] D.E. Adams and M. Nataraju. A nonlinear dynamical systems framework for structural diagnosis and prognosis. *International Journal of Engineering Science*, 40(17):1919–1941, 2002.

[16] D. Chelidze. Multimode damage tracking and failure prognosis in eletromechanical system. *Proceedings of SPIE - The International Society for Optical Engineering*, 4733:1–12, 2002.

[17] J. Luo, K.R. Pattipati, L. Qiao, and S. Chigusa. Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38(5):1156–1168, 2008.

[18] C. Hu, B.D. Youn, P. Wang, and Jo. T. Yoon. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering and System Safety*, 103:120–135, 2012.

[19] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *Proceedings of International Conference on Prognostics and Health Management*, 2008.

[20] M.J. Roemer C.S. Byington and T. Galie. Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. In *IEEE Aerospace Conference Proceedings*, 2002.

[21] M.R.J. Alves, C. de Oliveira Bizarria, and R.K.H. Galvão. Trend analysis for prognostics and health monitoring. In *Proceedings of Brazilian Symposium on Aerospace Engineering and Application*, 2009.

[22] M.G. Pecht. *Prognostics and health management of electronics*. John Wiley & Sons, New York, 2008.

[23] H.J. Lee and S.J. Roberts. On-line novelty detection using the kalman filter and extreme value theory. In *Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, 2008.

[24] S.J. Roberts. Novelty detection using extreme value statistics. In *IEEE Vision Image and Signal Processing*, pages 124–129, 1999.

[25] M. Schwabacher and K. Goebel. A survey of artificial intelligence for prognostics. *AAAI Fall Symposium - Technical Report*, pages 107–114, 2007.

[26] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[27] M. Schwabacher. A survey of data-driven prognostics. *AAAI Infotech at Aerospace Conference*, pages 1–4, 2005.

[28] C.J.C. Burges. A tutorial on support vector machine for pattern recognition. *Data Mining and Knowledge Discovery,*, 2:121–167, 1998.

[29] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: a survey. *ACM Computing Surveys (CSUR)*, 41(3):15:1–58, 2009.

[30] C.E. Rasmussen. *Gaussian processes for machine learning*. The MIT Press, 2006.

[31] C.K.I. Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. *Learning in Graphical Models*, pages 599–621, 1998.

[32] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research,*, 1:211–244, 2001.

[33] J. Cheng and R. Greiner. Learning bayesian belief network classifiers: algorithms and systems. In *Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, pages 141–151, 2001.

[34] R.B. Chinnam and P. Baruah. A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems. *International Journal of Material and Product Technology*, 20(1):166–179, 2004.

[35] B. Satish and N. Sarma. A fuzzy BP approach for diagnosis and prognosis of bearing faults in induction motors. In *IEEE Power Engineering Society General Meeting*, volume 3, pages 2291–2294, 2005.

[36] D.C. Swanson. A general prognostic tracking algorithm for predictive maintenance. In *IEEE Aerospace Conference*, volume 6, pages 2970–2977, 2001.

[37] C.S. Byington, M. Watson, and D. Edwards. Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *IEEE Aerospace Conference*, volume 6, pages 3581–3589, 2005.

[38] R. Huang, L. Xi, X. Li, C. Richard Liu, H. Qiu, and J. Lee. Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, 21(1):193–207, 2007.

[39] S. Du, J. Lv, and L. Xi. Degradation process prediction for rotational machinery based on hybrid intelligent model. *Robotics and Computer-Integrated Manufacturing*, 28(2):190–207, 2012.

[40] L. Peel. Data driven prognostics using a kalman filter ensemble of neural network models. In *International Conference on Prognostics and Health Management*, pages 1–8, 2008.

[41] J. Celaya, A. Saxena, S. Saha, and K. Goebel. Prognostics of power misfits under thermal stress accelerated ageing using data-driven and model-based methodologies. In *International Conference on Prognostics and Health Management*, 2011.

[42] C. Chen, G. Vachtsevanos, and M. Orchard. Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing*, 28:597–607, 2011.

[43] J. Liu, W. Wang, F. Ma, Y. Yang, and C. Yang. A data-model-fusion prognostic framework for dynamic system state forecasting. *Engineering Applications of Artificial Intelligence*, 25(4):814–823, 2012.

[44] J. Xu and L. Xu. Health management based on fusion prognostics for avionics systems. *Journal of Systems Engineering and Electronics*, 22(3):428–436, 2011.

[45] R. Orsagh, J. Sheldon, and C. Klenke. Prognostics/diagnostics for gas turbine engine bearings. In *IEEE Aerospace Conference*, pages 1165–1173, 2003.

[46] W. Hafsa, B. Chebel-Morello, C. Varnier, K. Medjaher, and N. Zerhouni. Prognostics of health status of multi-component systems with degradation interactions. In *6th IESM Conference*, 2015.

[47] L. Thomas. A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering and Systems Safety*, 16(4):297–309, 1986.

[48] G. Nguyen, P. Do, and A. Grall. Joint predictive maintenance and inventory strategy for multi-component systems using birnbaum's structural importance. *Reliability Engineering and System Safety*, 168:249–261, 2017.

[49] P. Ribot, Y. Pencole, and M. Combacau. Diagnosis and prognosis for the maintenance of complex systems. In *IEEE International Conference on Systems, Man and Cybernetics*, 2009.

[50] H. Zhang, M. Chen, and D. Zhou. Predicting remaining useful life for a multi-component system with public noise. In *Prognostics and System Health Management Conference*, 2016.

[51] D. Banjevic. Remaining useful life in theory and practice. *Metrika*, 69(2):337–49, 2009.

[52] T.D. Matteson. Airline experience with reliability-centered maintenance. *Nuclear Engineering and Design*, 89(2):385–390, 1985.

[53] H.A. Kinnison and T. Siddiqui. *Aviation Maintenance Management*. Tata McGraw Hill, New York, 2013.

[54] N. Papakostas, P. Papachatzakis, V. Xanthakis, D. Mourtzis, and G. Chryssolouris. An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48:604–612, 2010.

[55] A.K. Muchiri and K. Smit. Optimizing aircraft line maintenance through task re-clustering and interval de-escalation. In *Sustainable Research and Innovation Proceedings 3*, 2011.

[56] C. Sriram and A. Haghani. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A*, 37:29–48, 2003.

[57] C. Senturk and I. Ozkol. The effects of the use of single task-oriented maintenance concept and more accurate letter check alternatives on the reduction of scheduled maintenance downtime of aircraft. *International Journal of Mechanical Engineering and Robotics Research*, 7(2):189–196, 2018.

[58] H. Wang, J. Gao, and H. Wu. Direct maintenance cost prediction of civil aircraft. *Aircraft Engineering and Aerospace Technology*, 86(5):406–414, 2014.

[59] IATA. Airline cost management group (acmg) enhanced report. 2015.

[60] M. Baars. *Optimal Replacement Policy - Using Prognostics to Optimise Replacement in an Operational Environment*. Delft University of Technology - Faculty of Aerospace Engineering, Delft, 2018.

[61] M. Khwaja. *Strategies to reduce Maintenance cost*. Boeing lifecycle solutions, pg.32, 2012.

[62] Harbor Research Pervasive Internet Report. Approaching zero downtime: The center for intelligent maintenance systems (IMS)n. *Harbor Research Inc.*, pages 1–11, 2003.

[63] T. Grubic, L. Redding, T. Baines, and D. Julien. The adoption and use of diagnostic and prognostic technology within UK-based manufacturers. *The Journal of Engineering Manufacture*, 225(8):1457–1470, 2011.

[64] X. Lei and P.A. Sandborn. Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements. *Renewable Energy*, 118(Part B):188–198, 2018.

[65] F. Camci. Maintenance scheduling of geographically distributed assets with prognostics information. *European Journal of Operational Research*, 245(2):506–516, 2015.

[66] M. You and G. Meng. A predictive maintenance scheduling framework utilizing residual life prediction information. *Journal of Process Mechanical Engineering*, 227(3):185–197, 2012.

[67] F. Camci. System maintenance scheduling with prognostics information using generic algorithm. *IEEE Transactions on Reliability*, 58(3):539–552, 2009.

[68] Z. Li, J. Guo, and R. Zhou. Maintenance scheduling optimization based on reliability and prognostics information. In *Annual Reliability and Maintainability Symposium (RAMS)*, 2016.

[69] B. Zhang, L. Xu, Y. Chen, and A. Li. Remaining useful life based maintenance policy for deteriorating systems subject to continuous degradation and shock. In *51st CIRP Conference on Manufacturing Systems*, 2018.

[70] K. Bouvard, S. Artus, C. Berenguer, and V. Cocquempot. Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles. *Reliability Engineering and System Safety*, 96(6):601–610, 2011.

[71] B. Castanier, A. Grall, and C. Gerenguer. A condition-based maintenance policy with non-periodic inspections for a two-unit series system. *Reliability Engineering and System Safety*, 87(1):109–120, 2005.

[72] A. Van Horenbeek and L. Pintelon. A dynamic prognostic maintenance policy for multi-component systems. In *2nd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology*, 2012.

[73] T. Lin and C.Wang. A hybrid genetic algorithm to minimise the periodic preventive maintenance cost in a series-parallel system. *Journal of Intelligent Manufacturing*, 23(4):1225–1236, 2012.

[74] Z. Tian and H. Liao. Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering and System Safety*, 96(5):581–589, 2011.

[75] Y. Zhou, Z. Zhang, T.R. Lin, and L. Ma. Maintenance optimization of a multi-state series parallels system considering economic dependence and state-dependent inspection intervals. *Reliability Engineering and System Safety*, 111:248–259, 2013.

[76] K. Nguyen, P. Do, and A. Grall. Condition-based predictive maintenance for multi-component systems using importance measure and predictive information. *International Journal of Systems Science*, 1(4):228–245, 2014.

[77] K. Nguyen, P. Do, and A. Grall. Multi-level predictive maintenance for multi-component systems. *Reliability Engineering and System Safety*, 144:83–94, 2015.

[78] H. Vu, P. Do, A. Barros, and C. Barenguer. Maintenance grouping strategy for multi-component systems with dynamic contexts. *Reliability Engineering and System Safety*, 132:233–249, 2014.

[79] H. Vu, P. Do, and A. Barros. A stationary grouping maintenance strategy using mean residual life and the birnbaum importance measure for complex structures. *IEEE Transactions on Reliability*, 65(1):217–234, 2014.

[80] S. Wu, Y. Chen, Q. Wu, and Z. Wang. Linking component importance to optimisation of preventive maintenance policy. *Reliability Engineering and System Safety*, 146:26–32, 2016.

[81] N. Chalabi, M. Dahana, B. Beldjilali, and A. Neki. Optimisation of preventive maintenance grouping strategy for multi-component series systems: Particle swarm based approach. *Computers & Industrial Engineering*, 102:404–451, 2016.

[82] G. Niu and J. Jiang. Prognostic control-enhanced maintenance optimization for multi-component systems. *Reliability Engineering and System Safety*, 168:218–226, 2017.

[83] Z. Zhang, S. Wu, B. Li, and S. Lee. Optimal maintenance policy for multi-component systems under markovian environment changes. *Expert Systems with Applications*, 40:7391–7399, 2013.

[84] J. Hu, L. Zhang, and W. Liang. Opportunistic predictive maintenance for complex multi-component systems based on DBN-HAZOP model. *Reliability Engineering and System Safety*, 90:376–388, 2012.

[85] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research, Tenth Edition*. McGraw-Hill Education, New York, 2015.

[86] T. Zhang, Z. Cheng, Y. Liu, and B. Guo. Maintenance scheduling for multi-unit system: A stochastic petri-net and generic algorithm based approach. *Maintenance and Reliability*, 14(3):256–264, 2012.

[87] M. Marseguerra and E. Zio. Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation. *Reliability Engineering and System Safety*, 68:69–83, 2000.

[88] D. Chen and K.S. Trivedi. Optimization for condition-based maintenance with semi-markov decision process. *Reliability Engineering and System Safety*, 90(1):25–29, 2005.

[89] X. Yao, X. Xie, M.C. Fu, and S.I. Marcus. Optimal joint preventive maintenance and production policies. *Naval Research Logistics*, 52(7):668–681, 2005.

[90] E. Ijzermans. *Machine Learning for Predictive Maintenance; A Boeing 747 Bleed Air Valves case study*. Delft University of Technology - Faculty of Aerospace Engineering, Delft, 2018.