Engine Shop Visit Scheduling

A Reinforcement Learning optimization approach

Nícolas Winckler Musskopf

Delft University of Technology

TU Delft

Engine Shop Visit Scheduling

A Reinforcement Learning optimization approach

by

Nícolas Winckler Musskopf

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday 30^{th} January 2024.

Student number:	5626986
Project duration:	January 2023 – January 2024
Thesis committee:	Dr. B. F. Santos
	Dr. C. Falsetti
	Dr. M. L. Baptista
	Dr. M. Witteman
	I. Tseremoglou

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

A lot of people were essential for the conclusion of this research, without whom it would not be possible.

I would like to firstly thank my supervisor Dr. Bruno Santos, for all the support, patience and teachings along this journey. The help of Iordanis Tseremoglou was also essential, shedding light where I could not see any and providing me with a lot of the base that I didn't have at the start of this project. Furthermore, I would like to thank the entire aerospace faculty, all the staff, professors and lecturers that work hard to provide excellence.

My immense gratitude goes to the entire airline data science team at TUI Fly, who received me with open arms and were essential thought this process, providing me with all the necessary information, data, support and incredible experiences. My special gratitude goes to Marcelo Iannin and alumni Max Witteman, who actively participated in every step of this process and were always there for me, thank you!

I could not have gone through this entire process without the help of all my friends and my girlfriend Izabella Casagrande, who were not only there to celebrated with me through every achievement, but also got me back up at every bump in the way.

At last, but most importantly, I would like to thank my family. Without my family, my mom Ruth W. Musskopf, my dad Carlos Musskopf, my sister Tamara W. Musskopf, I could have never been here and experienced all the amazing things that I got to experience so far. Your support is, and always will be, the base for everything I do, and I will never be thankful enough. Talking about my family would not be completed without mentioning Mrs. Darlene Cox and Dr. Alan Cox, who have not only always cheered for me, but also actively participated in the making of this thesis since the start.

To each one of you, muito obrigado!

Nicolas Winckler Musskopf
Delft, January 2024

Contents

List of Figures

List of Tables

List of Abbreviations

ALS	Airworthiness Limitation
AoG	Aircraft on Ground
CBM	Condition-Based Maintenance
CM	Condition Monitoring / Corrective Maintenance
CMR	Certification Maintenance Requirements
DOZM	Date of Zero Margin
DP	Dynamic Programming
EASA	European Union Aviation Safety Agency
EGT	Engine Exhaust Temperature
EGTM	Engine Exhaust Temperature Margin
ESV	Engine Shop Visit
ETOPS	Extended-range Twin-engine Operational Performance Standards
FAA	Federal Aviation Administration
FBM	Failure-Based Maintenance
FJSP	Flexible Job-Shop Problem
FMEA	Failure Mode Effect Analysis
FOD	Foreign Object Damage
GA	Genetic Algorithms
HT	Hard Time
IFSD	In Flight Shutdown Rate
JSSP	Job-Shop Scheduling Problem
LLP	Life-limited Part
LP	Linear Programming
MAPS	Multi-Agent Scheduling Framework
MDP	Markov Decision Process
MILP	Mixed-Integer Linear Programming
MMDP	Multi-Agent Markov Decision Process
MPD	Maintenance Program Document
MRBR	Maintenance Review Board Report
MRO	Maintenance, Repair and Overhaul

MSG	Maintenance Steering Group
OC	On Condition
PM	Preventive Maintenance
RL	Reinforcement Learning
RUL	Remaining Useful Life
SSA	System Safety Analysis
SSM	State Space Model
TAT	Turn-around-time
TBM	Time-Based Maintenance
TCH	Type Certificate Holder

Introduction

The air transportation sector is an important factor in the world economy, as the airline industry alone spend in 2023 an amount corresponding to almost 1% of the globe's gross domestic product (GDP) [2]. This important sector of the economy, at the same time, operates on slim profit margins of less than 3%. In order to improve these results, operators are keen in finding ways to reduce costs and increase revenue, while keeping the safe records that aviation has established in recent times.

In order to maintain air transportation working smoothly and safely, all aircraft go through strict and extensive maintenance procedures. When considering aircraft engine maintenance, one of the most significant processes is the Engine Shop Visit (ESV), which is defined by [32] as:

> An engine removal, regardless of failure responsibility or maintenance category (scheduled or unscheduled), is classified as a shop visit whenever the subsequent engine maintenance performed prior to re-installation entails: a.) separation of pairs of major mating flanges, or b.) removal of a disk hub or spool.

In this context, the European airline TUI Fly, belonging to the leisure, travel and tourism company TUI Group, perceived the ESV scheduling process as a suitable point for operations optimization. The airline's data science department suggested the investigation of different approaches to this optimization problem. With access to experts in engine shop visit procedures, scheduling processes, data analysis, and machine learning, as well as to the airline's database, a deeper understanding of the process and possibilities was made viable.

In conjunction with the team, after mapping the current state of art for scheduling processes, it was decided to take an innovative approach of using reinforcement learning. Although reinforcement learning has been used to optimize the scheduling of many maintenance processes, it has never been used to schedule ESVs while considering the entire fleet of aircraft, pool of engines and the life-cycle of the engines in one formulation.

This thesis report is divided into two parts. In Part I, the scientific paper depicting the problem introduction and formulation, the solving methodology and the results is presented. Part II contains the relevant Literature Study that supports the research.

I

Scientific Paper

Engine Shop Visit Scheduling: A Reinforcement Learning Optimization Approach

Nicolas Winckler Musskopf,*

Delft University of Technology, Delft, The Netherlands

Abstract

The scheduling of engine shop visits quickly becomes a complex problem to solve as the number of aircraft and engines increases. In recent times, different approaches have been used to tackle this problem and optimize schedules, reducing costs and increasing revenue. This paper formulates the ESV scheduling problem as a Markov Decision Process and develops a reinforcement learning model that includes parameters such as engine performance and life limited parts status, maintenance constraints, and temporal factors. A prioritization algorithm is presented to optimize the learning process and allow for the scheduling of larger fleets. The results show a slight better performance in comparison to a greedy policy when evaluating aircraft availability, flexibility to the initial parameters and reduction in use of spare engines. On the other hand, the reinforcement learning provided lower scores and higher number of removals of aircraft from operations. In conclusion, the methodology proved that reinforcement learning is a viably way to optimize the ESV scheduling process, however a fine tuning of parameters might be necessary to approximate scores to a real revenue and cost relation, and reduce the number of aircraft interventions.

1 Introduction

Aircraft engines are complex and expensive systems, corresponding to up to 50% of an aircraft's price and 43% of an airline's Maintenance, Repair and Overhaul (MRO) costs [IATA, 2022]. Simultaneously, MRO costs correspond to about 11% of an airline's operational costs, thus reducing costs related to engine maintenance has the potential to generate a significant impact on the overall airline's operating costs.

For maintenance scheduling, three maintenance policies are mostly considered [Kumar et al., 2000], them being: Failure-Based Maintenance (FBM); Time-Based Maintenance (TBM); and Condition-Based Maintenance (CBM). In aviation, however, most items rely on the TBM policy [Kinnison and Siddiqui, 2013], where maintenance tasks are performed at fixed intervals based on the expected life-cycle of the part and by the operator's experience. TMBs might lead to an increase on unnecessary tasks and a consequential increase on costs with labor and parts, as the condition of the part is not taken into consideration. The CBM is a technique that tries to overcome the presented TBM disadvantages. It is based on diagnostics and prognostics, that is, on current observations of the system, using visual inspections, sensor monitoring among other methods, and estimates of the Remaining Useful Life (RUL) or of the failure probability at a determined future moment. In more recent years, prognostics has been further explored, encouraged by the growth in available data and data-driven methodologies to estimate future conditions. It reduces stub life, the waste of RUL for an item being replaced before its limit is reached, at the same time that avoids unexpected failures. It requires, however, that the item can be inspected, examined and/or monitored.

Aircraft are complex machines, and operators use the Maintenance Program Document (MPD) as a guide to understand the specific maintenance rules, schedules, and procedures for each part and system of the plane. This includes the engine subsystem, which consists of parts that each have their own maintenance requirements and intervals. This can lead to frequent interruptions in operation, as each component reaches its maintenance threshold. For minor maintenance tasks, the engines can stay attached to the wings, however more in-depth maintenance requires removing the engines and sending them to a Maintenance, Repair, and Overhaul (MRO) facility. There, a comprehensive engine inspection and repair process, known as an Engine Shop Visit (ESV), is performed.

Removing an engine from an aircraft will result in keeping the aircraft out of operations for the necessary time to remove the engine and replace it with a spare one. This process results in costs related to both the actual shop visit (SV) and the aircraft grounding, as well as also cause disruption in the airline's operations. This becomes an optimization process where the goal is to increase the use of engine's parts life to their maximum, and, simultaneously, reduce the number of times

*Msc Student, Sustainable Air Transport, Faculty of Aerospace Engineering, Delft University of Technology

an aircraft is removed from operation for maintenance tasks, while also reducing the number of spare engines required to retain the fleet in operation. In case of fleets with a low number of aircraft and engines, this type of optimization can be performed manually. In fact, many airlines still rely on manual solutions, using regular spreadsheets and personal experience from scheduling teams to obtain a feasible solution, which might be far from the most cost-efficient schedule [Lagos et al., 2020]. Newer optimization techniques, however, are being increasingly used by airlines to consider all factors involved in ESVs and reduce their total maintenance costs by more optimally scheduling their procedures, while also fulfilling all maintenance regulatory requirements.

Among the new optimization techniques, the use of reinforcement learning (RL) has been increasingly explored for maintenance scheduling of complex systems. In this paper, the ESV scheduling problem is formulated as a Markov Decision Process (MDP) and a novel approach using reinforcement learning for its optimization process is proposed, presented and tested. The methodology consists of creating a schedule by optimally defining what every engine will be doing, in the form of actions, at every time step for the evaluated period. The proposed model embrace a broader scope, considering not just the timing of ESVs but also the strategic allocation of engines across different aircraft, storage, and maintenance activities. Different workscopes of maintenance activities are also considered, where each workscope represents a different type of engine shop visit, with different effects on the engine's conditions after the action. The aim is to achieve a globally optimized solution for the fleet over the evaluation period, balancing factors such as operational efficiency, maintenance costs, and aircraft downtime.

In Section 2 a literature review from the main and most recent methods to produce and optimize ESVs schedules is presented. The problem is then presented, including its parameters, constraints and dynamics, in order to be mathematically formulated in Section 3. Section 4 presents the proposed methodology to solve the ESV scheduling problem in an optimal, or close to optimal, manner. The results for the proposed method are presented and discussed in Section 5, followed by the concluding remarks, as well as suggestions for future research, in Section 6.

2 Literature Review

When dealing with scheduling maintenance for complex machines, as for the case of aircraft engines, a general maintenance policy for the entire system is usually not applicable, as its different parts will have different inspection and replacement periods. This type of systems might lead to a higher number of disruptions in operations, as the process has to be stopped every time a part reaches its threshold. To solve this problem, one of the initial solutions was introduced through the concept of "opportunistic replacement", by [Jorgenson and Radner, 1960]. In it, a maintenance period required by one of the system's parts can be used as an opportunity to also replace or restore a secondary part that has not yet reached its end-of-life or has failed, thus avoiding a future operational disruption for that part's replacement. The optimal schedule will balance the trade-off between the waste of remaining useful life in the secondary part and the gain resulted from avoiding a higher downtime for the machine.

The ESV scheduling process began to be optimized with the development of operational research, especially with the use of linear programming. This type of problem requires many assumptions and simplifications in order to allow it to be mathematically modeled in a linear program or mixed-integer linear program (MILP), frequently implying in an unrealistic solution, as was first observed by [Boere, 1977]. In it, the author argues that not only the solution might be unrealistic, but the high unpredictability of an airline's operations might render a previously found optimal solution unfeasible. Their conclusion was that finding the optimal solution might be less valuable than finding a suboptimal solution, while good and robust. This was the base of many of the future research done in this field, aiming at providing good decision-support tools for the operators.

More recently, [Sun et al., 2023] created a decision-support tool with a discrete event simulation tool, using operational data such as engine deterioration status and degradation patterns, overhaul turn-around time and Life-Limited Parts (LLPs) to determine an initial set of possible ESV dates for a single engine's entire life-cycle span. LLPs are measured by the amount of flight cycles remaining up to its end-of-life point, where each flight cycle consist of a take-off proceeded by a landing. The engine's performance degradation status is measured by the engine's exhaust gas temperature margin (EGTM), that is, the margin between the maximum operational exhaust temperature for the engine, defined by the engine's manufacturer, and the actual measured exhaust temperature. In addition to the EGTM and LLP status, the authors also consider different engine parts that could provoke failures, estimating a random failure probability. A similar approach is seen in the research presented by [Mayordomo et al., 2010], using operational and contractual data as inputs for the prediction of when an ESV will be required, as well as estimate the maintenance costs. With this information, it generates different scenarios through an iterative algorithm, considering different thrust settings and shop visit workscopes. Both approaches aim at providing the operator's scheduling agent an easier method to consider all factors and estimate costs and end of life dates, without trying to optimize it.

Some approaches also explored the possibility of finding an optimized schedule by modeling the problem through MILP or MDP, and using a variety of optimization methods to solve it. In the case of [Almgren et al., 2012], the opportunistic replacement concept is used as base for its method, considering only the status of LLPs, the cost of the parts replacement and a maintenance base cost, which is irrespective of the workscope of the maintenance performed. Through six different constraints the problem is modeled, and new constraints are derived through Chvátal–Gomory rounding, so that the feasible region can be defined with a convex hull and, thus, an optimal solution can be obtained. The solution obtained with a commercial MILP solver achieved a 70% reduction in maintenance occasions and 34% reduction in total maintenance costs. Meanwhile, [Cai et al., 2016] considers both LLP status and EGTM degradation, modeling the LLP as having a normal distribution with very small variance. They explore the concept of just-in-time (JIT), where both engines may be restored while using only one spare engine and only three operational disruptions, by removing the second engine as soon as the first returns from its shop visit and swapping the spare engine's wing. The ideal moment to start this sequence of actions is determined using a mathematical model considering the maximum acceptable failure risk probability for the first engine and the maximum permissible probability of shortage for the second engine, where the permissible shortage probability is the probability that the second engine will require some type of maintenance while the first is still on its shop visit, thus forcing the use of an urgent spare engine, that might lead to higher leasing costs.

MDPs, however, may represent the problem more faithfully, as many of the problem's parameters are not linear and/or deterministic, as for example the failure probability, the EGTM degradation pattern and the number of flight cycles performed in a certain period. Through them, a state vector can then be used to represent the entire system, which can then be used to base the decision process. In [Li et al., 2019], the probability of failure of the engine's parts is correlated to the engine's performance, considering different LLPs and the EGTM, combining the optimization of maintenance intervals with the decision of the maintenance workscope for every shop visit opportunity. This is then optimized using reinforcement learning with a Gauss-Seidel value iteration algorithm, exploring different schedules with different workscopes, iterating over different policies in search of a sequence that reduces total maintenance costs, also considering the benefits of opportunistic replacement. The failure probability is also considered by [Camci, 2009], in conjunction with the inventory levels, focusing on these two parameters instead of the parts and performance threshold prognostics. In it, a binary string is used to represent the schedule, that is then used to calculate the total maintenance risk, where the maintenance risk is mathematically defined considering a sum of failure costs and losses due to maintenance processes. The optimization process uses genetic algorithms to iterate over different strings, seeking to reduce the total maintenance risk. When compared to the implementation of the FBM, TBM and CBM methods separately, using genetic algorithms to reduce maintenance risk provided a better result, showing the potential for alternative policies to the traditional methods. Similarly, [Hsu et al., 2010] also uses genetic algorithms, however using the chromosomes to present the number of flight cycles remaining for two life limited parts and the workscope of the corresponding shop visit, considering four different possible maintenance actions. In this case, the genetic algorithm is used to search for a solution where the total maintenance cost per flight cycle is reduced, thus also considering the impact of waste of remaining life when replacing an engine part. For these three methods, the optimal policy is obtained while considering a single engine.

Methods like genetic algorithms and reinforcement learning are also the state of art for the scheduling optimization of different maintenance processes, as is the case for aircraft maintenance, for example. This can be seen in the research of [van der Weide et al., 2022] and [Andrade et al., 2021], which try to solve the problem originally presented by [Deng et al., 2020] with genetic algorithms and reinforcement learning, respectively. [Deng et al., 2020] proposes a dynamic programming method to optimize the long-term planning of C checks to reduce the number of maintenance occasions and increase aircraft availability. [van der Weide et al., 2022] tries to reformulate this problem as an integer linear programming problem and solve it by using a chromosomic representation for each aircraft by displaying the number of time steps to its next maintenance procedure. Genetic algorithm is used, instead of a commercial MILP solver, due to the size of the problem. As for [Andrade et al., 2021], the original dynamic programming model is explored with a Deep Q-learning reinforcement learning process. Both methods reach better results then the original dynamic programming model. In particular, the reinforcement learning method, although requiring a long training process, is able to reach a solution for different initial scenarios within seconds, being thus flexible to produce a new schedule if unexpected operational situations arise. To the best of the authors knowledge, the use of reinforcement learning to achieve a flexible optimization policy for engine shop visit scheduling considering an entire fleet was not previously explored.

3 Problem formulation

The engine shop visit scheduling problem was defined through a problem introduction, the costs factors influencing the scheduling process, a mathematical formulation of the problems in terms of a MDP, including the definition of parameters, the action space, the state space the state transition functions and all constraints.

3.1 Problem introduction

The ESV scheduling optimization problem can be defined as follows: Considering the fleet of a major airline, where each aircraft requires a number Q of engines to be operational, how can operational costs be reduced by optimizing the schedule and workscope definition for all ESVs, considering all the operational and safety restrains?

Lets consider an entire fleet of aircraft, where each aircraft has Q engines. Once any of the engines reaches the performance deterioration threshold, in terms of EGTM, or one of its LLPs has reached the end of its useful life, defined in the MPD, it has to be removed from operations and be stored or sent to maintenance, where the part will be replaced or restored. Engines, however, can also be sent to a shop visit before reaching any threshold, if this is for some reason deemed beneficial, as in the case of opportunistic replacement for example. Furthermore, they can also undergo additional maintenance procedures, along with the restoration of the ESV triggering factor, through different shop visit workscopes. The scheduling must thus be defined by selecting the ESV dates and workscopes through a decision policy that takes into consideration all of the engine's parameters. A strategic decision policy that uses this information to reduce the maintenance costs will then generate an optimal schedule, which may then be introduced by the operator.

Engine shop visits can have multiple workscopes, where each is defined by a combination of parts that will be replaced or restored. Workscopes vary in size and complexity, as they could refer only to the parts related to the triggering factor, up to a complete overhaul, that is, for both performance and LLP items. Consequently, each workscope also has different costs and turn around times, in direct correlation to the procedure's complexity. A defined number of maintenance slots are available for simultaneous shop visits at any given time. Once an engine returns from a maintenance process, it can be reinstalled at the aircraft where it previously was, in a different aircraft, or left in storage, awaiting for a better installation opportunity.

Aircraft and engines are usually leased (or bought) together. The original aircraft from an engine is named its titled aircraft, and the corresponding engines to an aircraft are called its titled engines. At the end of an aircraft's leasing period, it has to be returned with its titled engines. Furthermore, engines might have other contractual requirements, as for example requiring a certain EGTM or a certain amount of remaining life on all LLPs from the engines.

Taking into account the ability to interchange engines between aircraft of the same fleet type, the possibility of synchronizing the grounding of several aircraft to facilitate this swap, and the fact that each engine exhibits unique patterns of degradation, operational expenses, and leasing agreements, it becomes evident that the complexity of the problem escalates rapidly. Noticeably, however, engines will rarely be directly swapped among aircraft in real operations, mostly going to a different aircraft only once it returns from a shop visit. Engines returning from a shop visit might also be stored, in case it is not advantageous to return it to its previous aircraft at that point of time, or used as a spare engine for another aircraft, in case no other spare engine is available.

In order to optimize the scheduling of ESVs, it is essential to understand the factors that trigger the necessity of one. Generally, five factors are their most common causes [Ackert, 2015]:

1. Expiration of a Life-Limited Part (LLP);
2. Performance deterioration;
3. Service bullet in compliance;
4. Emergency Airworthiness Directives;
5. Unexpected engine anomaly.

Items four and five, however, are reactive measures and cannot be predicted, thus not participating in the ESV scheduling process. Service bulletins generally do not require immediate action, giving the operator the flexibility to choose an appropriate time to do any necessary actions. The first two items, however, are the most frequent triggers and are the base for the long-term planning of ESVs. Performance deterioration is most commonly measured by the EGTM, thus requiring a data-driven CBM process. Life-Limited Parts, on the other hand, have defined number of flight cycles before they have to be replaced, requiring thus a version of the TBM policy.

3.2 Scheduling cost factors

During the optimization of the ESV schedule, the operator's main goal is to reduce costs to a minimum while maintaining safety levels and fulfilling all regulatory requirements. The main factors that will generate costs and have to be considered during the scheduling process are:

- **Operational costs:** Operational disturbances may be caused from both scheduled and unscheduled maintenance actions. If engines are allowed to operate closer to their operational threshold and their parts life limits, the operational reliability is reduced. Unscheduled maintenance actions generate a higher impact on the airline's operations, as flights might be canceled and/or delayed, generating costs in relocation, compensations and leasing of another aircraft. Scheduled maintenance will also impact the airline's operations, however the operator will have time to reschedule passengers and reroute airplanes, thus reducing extra costs. It is thus essential to schedule engine shop visits considering the impact of scheduled and unscheduled maintenance process, increasing the operational robustness to maintain the fleet available for an extended period of time.

- **Parts utilization:** The interval between scheduled ESVs determine the frequency that engine parts or components are replaced or maintained. By increasing the frequency of shop visits, the exploitation of the life time from parts can be increased, at the cost of also increasing the number of operational disturbances. Furthermore, the frequency of ESV also define the reliability of the engines. Extending the exploitation of the life time of parts can reduce the reliability of the engines and increase unscheduled maintenance costs.

- **Shop visit workscope decision:** When scheduling shop visits, airlines usually have a large number of different options on workscopes for the process. Engines might go through performance restoration processes, improving the EGTM, as well as replacing different life limited parts. Not all life limited parts on an engine have the same life span, meaning they might reach their limit at different time points. Shop visit costs are influenced by the number of tasks performed, as well as by fixed costs that do not vary with amount of tasks. Therefore, the cost per task per shop visit decreases as more tasks are undertaken at once. Using less broad workscopes will also lead to more frequent shop visits, which in turn will require for airplanes to be more frequently removed from regular operations. The airline has thus to decide if the reduction in costs per task, observed when performing a broader workscope, in conjunction with the reduction in the number of times the aircraft has to leave regular operations for an engine change, compensates for the wasted stub life on these parts that would not yet require replacement.

- **Contractual penalties:** The leasing of aircraft engines will very often come with obligations and penalties that will directly impact the ESV scheduling process, as well as the engine's life cycle maintenance costs. Lessors might, for example, require compensation if their engines performance are below a certain threshold. They might also require LLPs replacement if an engine goes to a shop visit for any another reason and it is close to its leasing end. These situations might lead to a high stub life, as they will be replaced regardless of how close they are to their life limit. As engines face a limited number of shop visits during their life cycles, the restrictions on an end of leasing shop visit will have an impact on the decisions taken in the previous shop visits, impacting their combined costs.

The scheduling of an ESV for one engine might also be affected by the scheduling of maintenance for other engines. Although the number of engines that can be at an ESV at the same time depends on the capacity of the MRO provider, if more engines are going through a shop visit than there are spare engines, some aircraft will have to stay out of operations, resulting in loss of possible revenue. On the other hand, buying or leasing more spare engines will also incur in high costs, thus a balance between the operational flexibility generated by extra spare engines and the costs that they generate has to be achieved.

The relation between different engines during the ESV scheduling process is also observed in other situations. As there is usually more then one engine installed on an aircraft, and the removal of an aircraft from operations incur operational costs, the decision of performing a shop visit in one engine might also create the opportunity of costs savings by performing a shop visit in the aircraft's other engine.

Lets consider, for example, that an aircraft is grounded for the removal and shipping of engine number one for an ESV, a spare engine will have to be installed on its place. Once the removed engine is returned from the shop visit, it is possible to reinstall engine number one and remove engine number two, replacing it with the same spare engine that was used as spare for engine one. This sequence is visualized in Figure 1. In this case, operational disruptions in the aircraft are reduced, as the aircraft only has to be removed from operations once to perform both the installation of engine one and the removal of engine two. The returned engine could also be installed on another aircraft, using it as a spare engine. In both cases, only one spare engine is used for the maintenance of two engines, being examples of opportunistic replacement that lead to reduced operational costs and reduction in the number of spare engines.

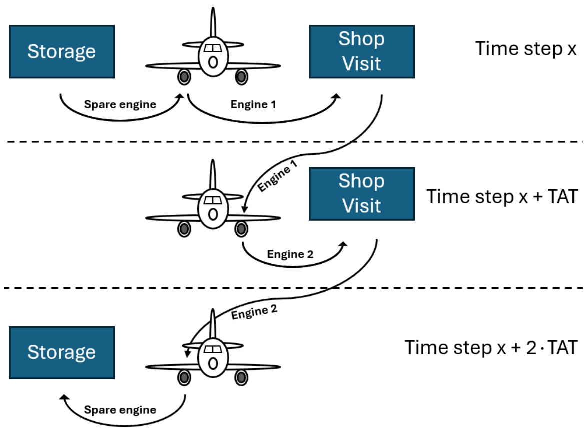


Figure 1: ESV opportunistic replacement sequence.

At last, flight regulations mandate that an aircraft undergoing simultaneous installation of two engines must undergo flight tests before resuming commercial service. This requirement does not apply when only one engine is replaced. These mandatory flight tests incur additional costs for crew and fuel and extend the period during which the aircraft is unavailable for commercial operations.

3.3 Formulation parameters

In this section the ESV scheduling problem is formulated using a finite-state and finite-action Markov decision process (MDP) framework, in order to be later easily implemented into a reinforcement learning algorithm.

3.3.1 Aircraft parameters

Aircraft require engines to be operational, and any engine may be assigned to any aircraft, as long as they are from the same type. Aircraft are represented by an index p, going from one to n, where n is the size of the fleet. The set of all aircraft is the set P.

The number of engines assigned to an aircraft p at any given moment t is represented by the engine counter variable $k_{p,t}$. Another aircraft parameter is the aircraft on ground (AOG) indicator $\Theta_{p,t}$. The AOG indicator is a binary variable where one indicates that, at the current time step, aircraft p is going through an engine removal or installation, so it has to be removed from regular operations.

3.3.2 Engine parameters

Each aircraft p has a number Q of engines. In this paper, Q is always equal to two. Engines are represented by the index e, going from the number one to the number m, where m is the total number of engines available. The combination of all titled engines and spare engines constitutes the set E. The titled aircraft of an engine is identified by the parameter z_e.

As each engine has a defined leasing period, the progress of the leasing is observed in variable $l_{e,t}$ as a proportional value in decimal representation, where zero indicated the starting time point of the leasing and one its end. The leasing progress variable is incremented by a value Δl_e every time there is a time step progression. The increment Δl_e corresponds to the ratio of one time step to the length of the leasing contract ι_e for engine e, as seen in Equations 1.

$$\Delta l_e = \frac{1}{\iota_e} \tag{1}$$

Engines have two parameters that define their deterioration status, one regarding LLPs $LS_{e,t}$ and one regarding EGTM $PS_{e,t}$. For LLPs, $LS_{e,t}$ indicates the state of the engine in terms of remaining useful life. The state of deterioration is represented by a proportional value in decimal representation, being zero the worst possible deterioration status, meaning zero flight cycles remaining on the critical LLP of the engine.

The deterioration of the LLP status $LS_{e,t}$ depends on the average number of flight cycles between two time steps γ. Aircraft do not perform the same number of flight cycles each day and, furthermore, as their schedule is only defined a few months before operations, it is not possible to determine the exact number of flight cycles that will be performed by the engine during its entire life-cycle, being possible only an estimation of it, using previous operations data. In this paper an average number of flight cycles between two time steps will be calculated and considered deterministic.

The average of flight cycles between two time steps can be calculated from the airline's data by evaluating the flight history of the airline, firstly calculating the average number of flight cycles between two time steps (υ) for the entire data set. For the purposes of this research, seasonal and yearly variations are not considered, being used a general average for the entirety of the flight history set. The average LLP status decay rate γ can then be calculated in terms of proportional value in decimal representation of the remaining useful life of a new LLP (LS_{max}), as indicated in Equation 2.

$$\gamma = \frac{\upsilon}{LS_{max}} \tag{2}$$

The performance status $PS_{e,t}$ is represented by the ratio between the EGTM at any given time step t to the maximum EGTM observed on the engine. This maximum EGTM is observed when the engine is new, when it will thus have the value of one, decreasing with time as the operation of the engine slowly deteriorates its internal parts, consequently reducing the performance efficiency until a EGTM of zero, where $PS_{e,t}$ will also be zero.

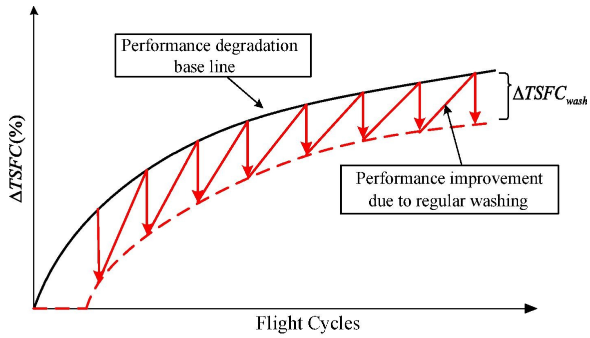


Figure 2: TSFC degradation and effect of engine wash for flight cycles.[Chen and Sun, 2018]

The exhaust gas temperature (EGT) is usually at its highest point during take-off, it is at this moment that the EGTM is recorded for each cycle. The EGTM decay presents a higher complexity problem, as many different factors impact the engine's EGTM degradation pattern. Among these factors are:

- **Flight environment:** The ingestion of suspended particles like dust, dirt, sand and others will provoke fouling and erosion of the engine's internal components, that will have as effect the deterioration of the engine's performance [Giorgi et al., 2018]. Consequently, flying in conditions with a higher concentration of particles, as for example in desert conditions [Ryder et al., 2023], will increase the rate of degradation.
- **External temperature:** The intake air temperature significantly affects the EGTM in aircraft engines, as a higher intake temperature leads to an increase in the overall temperature during the engine's operating cycle, accelerating wear and tear of internal parts and thus, increasing the deterioration rate.
- **Flight settings:** Factors such as take-off runway length, weather conditions and airline's strategy (through cost index) will impact on the configuration of the engine for each take-off and its flight performance. Furthermore, engines of the same type can be configured to different thrust settings. Changing the thrust setting requires small interventions in the engine that cannot be performed for each flight. Increase in thrust

is accompanied by increase in internal temperatures and pressures, factors that increase internal degradation and, consequently, performance deterioration. Engines with a higher thrust setting and operating more often in constrained environments (with shorter runways, higher airport altitudes, higher take off temperatures, etc) will thus have a higher deterioration rate.

- **Engine Washes:** The frequency of engine washes will directly influence the EGTM, as the cleaning process will reduce the effect of fouling in the engine. Scheduling them optimally, however, presents a challenge of its own. When an engine accumulates a high number of flight cycles, the impact of erosion becomes significant enough that an engine wash alone is no longer sufficient to substantially improve its performance [Chen and Sun, 2018]. Figure 2 exemplifies this pattern with the Thrust Specific Fuel Consumption (TSFC), that can be directly correlated to EGTM [Chen and Sun, 2018].

- **Engine age:** The degradation pattern is in reality not linear, having a higher rate at the beginning of the engine's life and a lower rate towards its end, as can be also seen in Figure 2. For the purpose of this research, this effect is disregarded.

The engine degradation is consequently a stochastic process when evaluating its progression over each cycle. It will, however, have a long term rate that is measurable and uniform among the fleet, as long as the entire fleet faces similar engine wash policies and operational environments. This assumption can be observed as realistic when plotting real EGTM data from a major European airline, as seen in Figure 3, where the short term impact of engine washes and shop visits are clearly visible, as well as the long term deterioration pattern between shop visits. As the ESV scheduling process is a long-term problem, the short-term fluctuations can be ignored and the EGTM deterioration per time step can be considered deterministic. The airline has to model this parameter in order to more precisely represent the pattern seen by its operations.

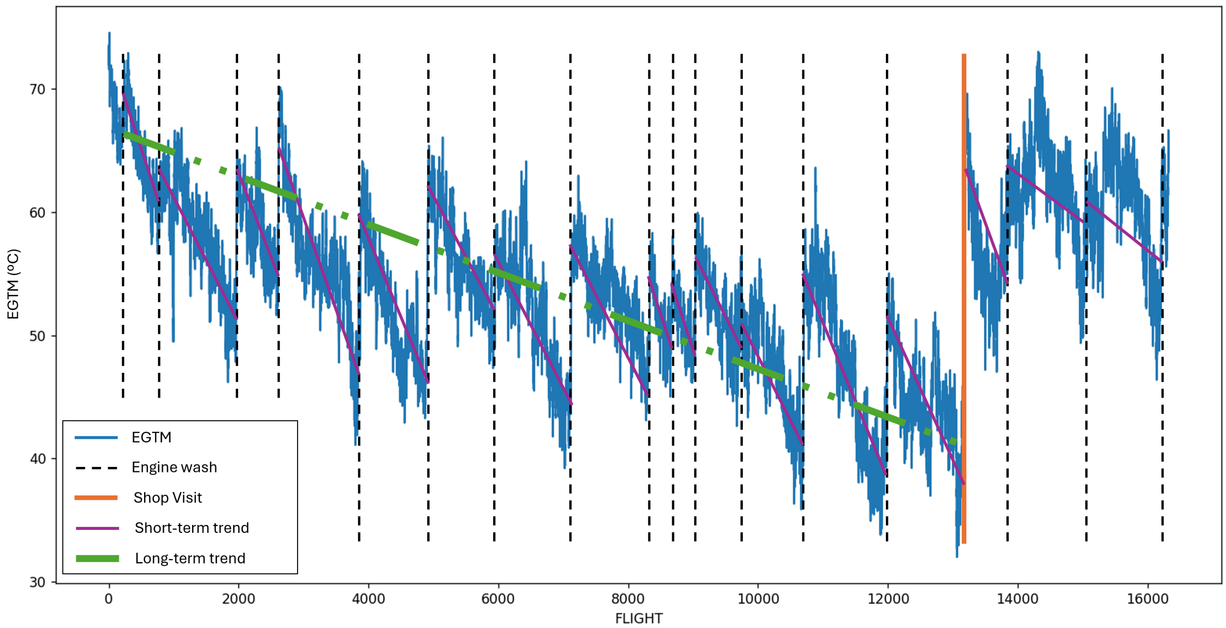


Figure 3: Real life example of EGTM deterioration with flight cycles.

The EGTM decay per time step σ is thus a variable that depends on each airline's historical data and will be unique for each operator. It can be calculated by firstly using a linear interpolation of the EGTM for each engine for the period between shop visits, and then calculating the average of rates ζ for all engines of the same type. σ can then be be calculated as a proportion from the average EGTM decay per time step to the maximum EGTM for a new engine PS_{max}, as calculated in Equation 3.

$$\sigma = \frac{\zeta \cdot \upsilon}{\max(PS_{max})} \tag{3}$$

The average rate of decay for both LLP and EGTM deterioration statuses, $\overline{\lambda}_{e,t}$ and $\overline{\xi}_{e,t}$ respectively, are also computed considering the last 5 time steps. They are responsible to represent the pattern of deterioration of any engine at any given time.

From the LLP and EGTM deterioration statuses, a prediction for each engine's future critical status is represented by variable $CS_{e,t}$. The factor with the highest degradation will be considered as critical and a linear interpolation is performed to predict the future state of this value. The objective of this variable is to allow, during the scheduling process, to evaluate if an engine will likely reach one of its thresholds during the summer period, which would require temporarily removing the aircraft from operation for the engine replacement. As the summer period is an interval of high demand, removing an aircraft from operation to send an engine for a shop visit during this time will have a larger impact on revenue, in comparison to doing so on other periods.

The airline also has to define at which degradation point it wants to stop the engine from staying in operations. As the real deterioration per cycle is stochastic, it might be desired to limit the minimum predicted degradation to a value slightly higher than the regulations minimum. This is done in order to avoid unscheduled removals, caused by an engine that had a higher than expected decay. This limit is defined by parameters LS_{min} and PS_{min}, for LLP and EGTM statuses respectively.

3.3.3 Temporal parameters

A few parameters depend only on time, where the current time step is indicated by the variable t. For example, airlines have agreements with MRO providers that define the maximum number of engines that may go to a shop visit at the same time. Based on the number of engines assigned to an ESV in the current time step, the available capacity variable $M_{e,t}$ is calculated and expressed as a proportional value, being zero maximum capacity and one no available slots.

As the summer period is a cyclical period that is of importance during the scheduling process, two variables indicate the relation of the current time step t to the summer period. The summer approach variable τ_t indicates how close t is to the start of the next summer period, being a proportional value where zero indicates the first time step after the summer period ending point, and one the beginning of summer. $\Delta\tau$ corresponds to the increase in one time step in relation to the amount of time steps between the end of a summer period t_{end_1} and the beginning of the next one t_{start_2}, as calculated in Equation 4.

$$\Delta\tau = \frac{1}{t_{start_2} - t_{end_1}} \tag{4}$$

The summer progress variable ρ_t indicates how close t is to the summer end, once it has started. It is a percentage value that stays null for most of the period, linearly increasing from zero to one during the summer period and returning to 0 once it has ended. $\Delta\rho$ corresponds to the increase in one time step in relation to the length of a summer period Δt_s, as calculated in Equation 5.

$$\Delta\rho = \frac{1}{\Delta t_s} \tag{5}$$

The airline also has to define a few constant parameters. Firstly, is has to evaluate, based on their contracts and historical data, what is the average turn around time ϕ required to receive back an engine, once it has been sent to a shop visit. The airline also has to define the point at which the engine degradation prediction has to be evaluated. This is indicated in terms of the summer approach variable and presented as τ_{SV}. At this point, engines that are predicted to trigger the necessity of an ESV during the summer have to be replaced, in order to avoid summer disruptions.

3.3.4 Transitional parameters

A few parameters must be calculated only for the transition of states for certain actions. These parameters are not explicitly presented in the state space, but have an impact on how it is updated.

As observable effect of an engine that undergoes an ESV with a workscope focused on performance restoration is its EGTM increase. A shop visit, however, will not fully recover the engine's performance to the same level as from a new one. The performance status of the engine after shop visit will thus not be the maximum value of one, rather being a slightly lower value.

This recovered status value R_{PS} has to be stochastically modeled using the data of EGTM values observed right after shop visits. As the EGTM is a product of many different factors, evaluating what is the exact impact of the shop visit only based on the EGTM observed directly after it will lead to misleading results. Consequently, it is necessary to use a sequence of data points after an ESV.

Once an average of EGTM after a shop visit is determined for enough shop visits within the same engine type group, it is necessary to determine the statistical model that best fit the results distribution. Then, an estimation method can be used to estimate the model parameters. For this paper, based on data from a major European airline, the recovered status R_{PS} was expressed following a normal distribution with a low standard deviation, as seen in Equation 6.

$$R_{PS} \sim \mathcal{N}(\mu, \sigma^2) \tag{6}$$

This recovery percentage is applied cumulatively to the last status after a shop visit $PS_{e,t-y}$, where y is the number of time steps since the last shop visit. If the engine has not yet undergone a shop visit, $PS_{e,t-y}$ is considered as one. Thus, the performance status after the shop visit being applied at the moment can be calculated by Equation 7.

$$PS_{e,t+\phi} = PS_{e,t-y} \cdot R_{PS} \tag{7}$$

The performance recovery rate r_{p_e} is based on the expected performance status at the end of the maintenance process, using the recovery probability distribution previously defined, and the current performance status. This rate calculation is necessary to correctly increase the engine's performance status up to the recovered status in the amount of time steps defined by the turn around time ϕ.

On the other hand, the LLP status recovery depends upon the number of workscopes available for LLPs replacements. If there is more than one, as for example by only replacing some of the parts, then the status after the shop visit will represent the smallest RUL of the parts that were not replaced. For this paper only one type of workscope limited to LLPs is considered, where the LLP status will be fully restored every time the parts are replaced, as expresses by Equation 8.

$$LS_{e,t+\phi} = 1 \tag{8}$$

As the LLP condition when replaced is deterministic, the LLP recovery rate r_{l_e} can be defined by the ratio between the difference of one and the last LLP status $LS_{e,t}$ before the start of the ESV and the turn around time ϕ.

3.4 Action Space

The action space is defined by all possibilities of actions for an engine. An engine might be in operation in an aircraft, in storage or in one of the possible shop visit workscopes. Each aircraft has one corresponding action in the action space. The number of possible actions is thus a variable, depending on the number of aircraft in the environment and the number of different shop visits that can be performed. LLP parts in an engine may have different life spans and, consequently, there could be a number of different workscopes, one for each set of parts that can be replaced, as well as the possible combination of sets. For this paper, however, only three different shop visit workscopes are considered, they are respectively:

1. **LLP Shop Visit (LPV):** In this action, only the deterioration state correspondent to the LLP status $LS_{e,t}$ is restored. This action is analogous to a shop visit which is limited to the replacement of all LLP parts.

2. **Performance Shop Visit (EPV):** In this action, only the deterioration value correspondent to the EGTM status $LP_{e,t}$ is restored. This action is analogous to a shop visit which is limited to the procedures for restoration of engine performance.

3. **Pair Shop Visit (PSV):** In this action, the degradation values for both LLPs and EGTM status are recovered. This action is analogous to a complete shop visit.

Actions are enumerated from the first aircraft to the last one, followed by the storage action, LLP SV, Performance SV and, at last, Pair SV. Actions 1 to n are considered “operational actions” and constitute the set of actions O, where “n" is the number of aircraft on set P. Storage actions, in this case represented only by action $n+1$, constitute the set of actions S. At last, LPVs, EPVs and PSVs are shop visit actions and constitute the set of actions F. The taken action is represented by variable $a_{e,t}$. It is an integer such that $1 \leq a_{e,t} \leq n+4$.

The action space can be visualized by Figure 4.

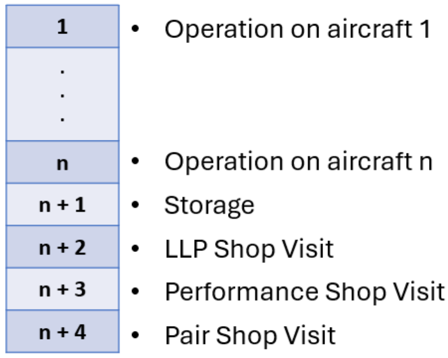


Figure 4: Numeric formulation for action space.

3.5 State Space

The combination of the statuses for the fleet of aircraft and pool of engines, in conjunction with the temporal parameters, represent the state of the scheduling environment at any given moment. A final schedule has this combination of parameters defined for all time steps in the analyzed period.

Parameters are divided into two types, depending on at which transition point they are updated in the state space. Variables of ”Type 1" are updated at each decision step. On the other hand, variables of ”Type 2" are updated only when there is progression from one time step to the following.

Each engine e on the set of engines is defined by a state vector $s^E_{e,t}$, where all variables combined indicate the status of the engine at each time step t, such as:

$$\begin{aligned} s^E_{e,t} = &[\underbrace{l_{e,t}, CS_{e,t}, a_{e,t-1}, \overline{\lambda}_{e,t}, \overline{\xi}_{e,t}, LS_{e,t}, PS_{e,t},}_{\text{Type 2}} \\ &\underbrace{a_{e,t}}_{\text{Type 1}}, z_e] \end{aligned} \tag{9}$$

Each aircraft in the operator’s fleet is also defined by an aircraft state vector $s^P_{p,t}$ at every time step t, containing the engine counter $k_{p,t}$ and the AOG indicator $\Theta_{p,t}$, such that:

$$s^P_{p,t} = [\underbrace{k_{p,t}, \Theta_{p,t}}_{\text{Type 1}}] \tag{10}$$

Three variables are only time dependent, constituting the time state vector s^T_t. They are: summer approach τ_t, summer progress ρ_t and MRO available capacity $M_{e,t}$.

$$s^T_t = [\underbrace{\tau_t, \rho_t}_{\text{Type 2}}, \underbrace{M_{e,t}}_{\text{Type 1}}] \tag{11}$$

The state space vector s_t can then be defined by the concatenation of the engine state $s^E_{e,t}$ for all engines at time t, aircraft state $s^P_{p,t}$ for all aircraft at time t and time state s^T_t at time t, as seen in Equation 12.

$$s_t = \left[s^E_{1,t}, s^E_{2,t}, \ldots, s^E_{e,t}, s^P_{1,t}, s^P_{2,t}, \ldots, s^P_{p,t}, s^T_t\right] \quad (12)$$

All sets required for the definition of the environment are presented by Table 16 in Appendix A. All individual parameters are presented by Table 15 in Appendix A.

3.6 State transition

The application of an action $a_{e,t}$ to an engine e might have different consequences in the state space, depending on the type of action. Once an action is chosen, the new state space is defined through a state transition function. The state space is updated every time that an action is chosen for one of the engines.

The decision-making process in which a single action is selected for each engine at every time step introduces a new layer of complexity. While the number of potential actions remains fixed at each step, the impact of a selected action on the state space is dependent upon the engine that is receiving that action. This "active engine", however, is not explicitly observable in the state space. It is always represented by the leftmost action with a null value inside the engine state part of the state space.

"If the action $a_{e,t}$ for an engine is operational, or if the preceding action $a_{e,t-1}$ was operational and the current action $a_{e,t}$ is not, the aircraft where the engine currently operates (or was last operational) is designated as the active aircraft α_p."

3.6.1 Type 1 parameters

Type one parameters require an update after each decision step. Firstly, as soon as an action is decided for an engine e, the action variable $a_{e,t}$ receives its value. Once $a_{e,t}$ is updated, the engine counter is also updated for the active aircraft α_p, as seen in Equation 13.

$$k_{\alpha_p,t} = \sum_{i \in E} \mathbb{I}\left(a_{i,t} = \alpha_p\right) \quad (13)$$

The AOG indicator $\Theta_{p,t}$ is set as one for the active aircraft α_p if the active engine was previously not in operation and at the current time step is. In this case the active aircraft is also included into the set of aircraft with engine installations Li_t for the current time step t. The AOG indicator will also be set as one if the engine was in operation at the previous time step and is not at the current one. In this case the engine is included into the set of aircraft with engine removals Lr_t. The value of $\Theta_{p,t}$ can thus be represented by:

$$\Theta_{\alpha_p,t} = \begin{cases} 1 & \text{if } a_{e,t-1} \leq n \text{ and } a_{\alpha_e,t} > n, \\ 1 & \text{if } a_{e,t-1} > n \text{ and } a_{\alpha_e,t} \leq n, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

The MRO available capacity $M_{e,t}$ is calculated after every step as for Equation 15.

$$M_{e,t} = \frac{1}{M_{max}} \sum_{e \in E} \mathbb{I}(a_{e,t} \in F) \quad (15)$$

where M_{max} is the MRO provider maximum capacity.

Once all type one parameters go through their transition functions, the active engine is interchanged to the next engine in the set of engines E, up to the last engine, of number m. In case the active engine e is the last engine, all type two parameters go through their state transition function.

3.6.2 Type 2 parameters

Type two variables are only updated when there is a change on the time step t. The leasing progress $l_{e,t}$, the summer approach τ_t, and the summer progress ρ_t are independent from actions assigned to the engines.

$l_{e,t}$ is recalculated for every engine e taking into consideration its previous value and the time step increase Δl_e, as seen in Equation 16.

$$l_{e,t+1} = l_{e,t} + \Delta l_e \quad (16)$$

The summer approach τ_t, and the summer progress ρ_t depend only on their previous status. If τ_t indicates that summer has not yet started, then it is increased by the amount $\Delta\tau$. Else, it remains at the same value of one, indicating that the current time step is in a summer period, unless the summer progress variable indicates the end of the summer at the current time step, in which case the summer approach variable is reset to zero:

$$\tau_{t+1} = \begin{cases} \tau_t & \text{if } \tau_t = 1 \text{ and } \rho_t \neq 1, \\ 0 & \text{if } \tau_t = 1 \text{ and } \rho_t = 1, \\ \tau_t + \Delta\tau & \text{otherwise.} \end{cases} \quad (17)$$

Similarly, the summer progress ρ_t is zero up to the point where the variable τ_t indicated the start of summer, increasing in value by $\Delta\rho$ up to the end of the summer period, when it is reset to 0:

$$\rho_{t+1} = \begin{cases} \rho_t + \Delta\rho & \text{if } \tau_t = 1 \text{ and } \rho_t \neq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

The remaining type two variables depend on the taken action. The parameter $a_{e,t-1}$ is updated with the action taken in $a_{e,t}$. Action variable $a_{e,t+1}$ is reset to zero for all engines e.

In case the previous action $a_{e,t-1}$ is a shop visit action, and it is the initial time step for this task, the recovery rate r_{p_e} has to be calculated, as seen in Equation 19.

$$r_{p_e} = \frac{PS_{e,t+\phi} - LS_{e,t}}{\phi} \quad (19)$$

For the case where only one workscope is available for LLPs, thus representing the replacement of all LLPs in the engine, the LLP recovery rate r_{l_e} can be calculated via:

$$r_{l_e} = \frac{1 - LS_{e_t}}{\phi} \tag{20}$$

The deterioration statuses $LS_{e,t}$ and $PS_{e,t}$ can then be calculated as represented in Equations 21 and 22, respectively.

$$LS_{e,t+1} = \begin{cases} LS_{e,t} - \gamma & \text{if } a_{e,t} \leq n, \\ LS_{e,t} & \text{if } a_{e,t} = n+1 \text{ or } a_{e,t} = n+3, \\ LS_{e,t} + r_{l_e} & \text{otherwise.} \end{cases} \tag{21}$$

$$PS_{e,t+1} = \begin{cases} LS_{e,t} - \sigma & \text{if } a_{e,t} \leq n, \\ LS_{e,t} & \text{if } a_{e,t} = n+1 \text{ or } a_{e,t} = n+2, \\ LS_{e,t} + r_{l_e} & \text{otherwise.} \end{cases} \tag{22}$$

The average rates of decay $\overline{\lambda}_{e,t+1}$ and $\overline{\xi}_{e,t+1}$ can then be calculated considering the last 5 time steps, as seen in Equations 23 and 24 respectively.

$$\overline{\lambda}_{e,t+1} = \frac{1}{5} \sum_{i=0}^{4} (LS_{e,t-i} - LS_{e,t-i-1}) \tag{23}$$

$$\overline{\xi}_{e,t+1} = \frac{1}{5} \sum_{i=0}^{4} (PS_{e,t-i} - PS_{e,t-i-1}) \tag{24}$$

The critical engine prediction is also calculated. As the intention of this variable is to estimate if an engine will have to be removed during a summer period, the interpolation is calculated for a period corresponding to the total summer length Δt_s plus the turn around time ϕ, so that even if it goes to a shop visit, it can be reinstalled before the beginning of summer. This can then be calculated through Equation 25.

$$\begin{aligned} CS_{e,t+1} = \min[&LS_{e,t} - (\phi + \Delta t_s) \cdot \gamma, \\ &PS_{e,t} - (\phi + \Delta t_s) \cdot \sigma] \end{aligned} \tag{25}$$

The sets Li_t and Lr_t also have to be restarted at each new time step so that they are empty (as no installations or removals were yet performed in the following time step), so that:

$$Li_{t+1} = \{\} \quad \text{and} \quad Lr_t = \{\} \tag{26}$$

Once all type two parameters are updated, the current engine index e in analysis is set back to one, as it corresponds to the first engine in the engine set E. The current time step t is increased by one. This process is repeated up to the last engine at the last time step. The implicit consequence of this process sequence is that an action decision step does not always correspond to a time step.

3.7 Reward Function

The reward function was formulated with the goal of accurately representing how the airline perceives the problem. Shop visits are beneficial in the long term, as they are necessary to keep aircraft operational (generating revenue), while producing high costs in the short term, when they are performed. This dynamic has to be reproduced by the reward function, so that any formulated schedule has a realistic score representation.

The main source of income for an airline is the operation of its aircraft. This is represented in the reward function through a positive reward Rv, that is generated every time an aircraft is in operation. An aircraft is considered in operation at a time step t whenever two engines are assigned to it, so that:

$$a_{e,t} \in O \implies Rv = \begin{cases} 1, & \text{if } k_{\alpha_p,t} = 2 \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

The result of equation 27 is that a positive reward is granted exclusively when assigning the second engine to an aircraft, whereas the first engine receives no reward. To prevent excessive complexity in the fleet mix, a minor penalty is imposed on an operational aircraft if one of engines used for its operation is not on its titled aircraft. This penalty is represented by Equation 28.

$$k_{\alpha_p,t} = 2 \implies Pn_t = \begin{cases} 0.00, & \text{if } (a_{e_1,t} = z_{e_1}) \land \\ & \qquad (a_{e,t} = z_e) \\ -0.05, & \text{otherwise} \end{cases} \tag{28}$$

where e_1 is the other engine also assigned to the aircraft α_p.

Although ESV are desired, they are costly operations. Not imposing any penalty on shop visits would result in an excessive number of them. Therefore, it is necessary to assign a negative reward to shop visits, as indicated in Equation 29. The variables c_l, c_e and c_p represent the penalty for each possible shop visit action. These variables have to be defined by the operator to be proportional to the real revenue and cost values observed by the airline.

$$a_{e,t} > n+1 \implies Pn_{sv} = \begin{cases} -c_l, & \text{if } a_{t,e} = n+2 \\ -c_e, & \text{if } a_{t,e} = n+3 \\ -c_p, & \text{if } a_{t,e} = n+4 \end{cases} \tag{29}$$

The act of installing or removing an engine generates a loss of revenue in form of removing the corresponding aircraft from normal operations. However, as the reward is calculated for each engine and not for each aircraft, and to allow for the exploration of opportunistic replacement, only the first engine that requires the removal of the aircraft from the roster should be penalized. These operations have a higher revenue impact in case they occur during summer, so the penalty for AOGs during summer are doubled, so that:

$$\Theta_{\alpha_p,t} = 1 \implies Pn_{AOG} = \begin{cases} 0.0, & \text{if } (p \in Li_t) \wedge (p \in Lr_t) \\ -c_a, & \text{else if } \tau_t \neq 1 \\ -c_s, & \text{otherwise} \end{cases} \quad (30)$$

Variables c_a and c_s represent the cost of grounding an aircraft and doing so in summer, respectively. As was the case for c_l, c_e and c_p, they are proportional costs to the revenue generated by an aircraft during one time step progression and have to be defined by the operator.

The final reward for the step in execution is the sum of the reward with all penalties, as in Equation 31.

$$Rw_{e,t} = Rv + Pn_t + Pn_{sv} + Pn_{summer} + Pn_{AOG} \quad (31)$$

and the total score for the episode is the sum of the rewards for all engines at all time steps. An episode is an entire scheduling process, from the first action decision for the first engine at the first time step up to the last engine at the last time step. Equation 32 presents the equation for the episode's score.

$$TS = \sum_{t \in T} \sum_{e \in E} Rw_{e,t} \quad (32)$$

3.8 Constraints

If at each decision point all actions from the action space are available, regardless of the current state space, future state spaces could represent scenarios that are not possible in real life, thus being of no interest in the scheduling process. To maintain any defined schedule within the bounds of real-world feasibility, a set of eleven constraints was defined. These constraints are applied by selectively removing infeasible or restricted actions from the set of possible actions available at each decision point. By doing so, it is ensured that the resulting action space reflects realistic operational boundaries and practical maintenance considerations.

3.8.1 Minimum deterioration threshold

This constraint is introduced to enhance the model's alignment with realistic operational scenarios. While in a real-world scenario the decision of sending an engine with low deterioration statuses to a shop visit is possible, it would provoke a very high stub life. This type of decision is thus unlikely and will almost certainly be far from an optimal schedule. Constraining shop visit actions from engines that are above a certain deterioration threshold $LS_{\min}$ and $PS_{\min}$ will in turn simplify the decision-making landscape, focusing the model's attention on more imminent and critical maintenance decisions. This constraint can be seen by Equation 33.

$$(LS_{e,t} \geqslant LS_{\min}) \wedge (PS_{e,t} \geqslant PS_{\min}) \implies a_{e,t} \leqslant n+1 \quad (33)$$

3.8.2 Maximum degradation for operations

While the engine could stay in operations up to the moment it reaches its threshold for LLP flight cycles or EGTM, operators will frequently desire to restrain engines from getting to close to these values, as it decreases the engine's reliability. Operators must therefore decide what is the maximum degradation statuses for performance PS^{max} and LLP LS^{max}, at which point engines have to be removed from operations, as seen by Equation 34.

$$(LS_{e,t} \leqslant LS^{max} \wedge PS_{e,t} \leqslant PS^{max}) \implies a_{e,t} \geqslant n+1 \quad (34)$$

3.8.3 MRO provider capacity

Once the number of engines sent to a shop visit equals the total amount of available slots agreed between the airline and the provider, engines are restrained to stay in operation or stay in storage, as seen in Equation 35.

$$M_{e,t} = 1 \implies a_{e,t} \leqslant n+1 \quad (35)$$

3.8.4 Turn around time constraint

Engines sent to a shop visit will stay at the MRO provider until they are fit to be returned to the airline. During this time, up to its end, no other actions should be available for the agent. This is represented by Equation 36.

$$(a_{e,t-1} \geqslant n+1) \wedge \left(\sum_{i=1}^{\phi} \mathbb{I}\left(a_{t-i,e} > n+1\right) \geqslant \phi \right) \implies a_{t,e} = a_{t-1,e} \quad (36)$$

3.8.5 Number of engines constraint

The number of engines assigned to any given aircraft must not exceed the specified quantity that the aircraft is designed to accommodate, denoted as Q. This results in constraining the corresponding actions for all aircraft that already have Q engines assigned to them, as an extra installation is impossible. This constraint is represented by Equation 37.

$$k_{p,t} = Q \implies a_{t,e} \neq p \quad \forall p \in P \quad (37)$$

3.8.6 Summer constraint

Airlines will frequently avoid performing heavy maintenance on its aircraft and engines during the peak demand seasons, particularly during summer, as these are the most profitable periods. To represent that, only engines that were in storage at the previous time step may have a shop visit action assigned to them during this period. Equation 38 represents the summer constraint.

$$(a_{e,t-1} < n+1) \land (\tau_t = 1) \implies a_{e,t} \leqslant n+1 \tag{38}$$

3.8.7 Pre-summer prediction

If it is predicted that an engine will require a shop visit during the summer period as a result of reaching its operational threshold, the engine has to be removed and sent to storage or maintenance before the summer begins, avoiding a removal and installation process during the peak-season period. Equation 39 represents how this constraint is implemented in the environment.

$$\begin{aligned}(\tau_t = \tau_{SV}) \land (CS_{e,t} \leqslant \max[LS_{min}, PS_{min}])\\ \implies a_{e,t} \geqslant n+1\end{aligned} \tag{39}$$

3.8.8 Titled aircraft constraint

As previously explained in Subsection 3.1, engines are most frequently leased together with an aircraft and will have to be returned with that same aircraft once its leasing reaches its contractual end. It is thus necessary to guarantee that the engine is on its titled aircraft once it gets close to the point where it has to be returned. The maximum date on which an engine might be out of its titled aircraft is defined by the operator, and is seen in Equation 40 by the parameter l_{limit}.

$$(l_e \geqslant l_{limit}) \implies a_{e,t} \geqslant n+1 \text{ or } a_{e,t} = z_e \tag{40}$$

3.8.9 Leasing end constraint

Once an engine reaches the end of its leasing period, it has to be returned to the lessor and cannot be further used by the airline. This is reproduced by restraining the engine to stay in storage once its leasing progress variable reaches the maximum value of one, symbolizing the end of the leasing, as seen in Equation 41.

$$(l_e \geqslant 1) \implies a_{e,t} = n+1 \tag{41}$$

3.8.10 Double installation constraint

To avoid the additional costs that installing engines at the same time generate, as stated in Section 3.2, it is beneficial to restrict the amount of engines installed in an aircraft to one at a time.

This is done by evaluating if the engine was in storage or maintenance at the previous time step. If it was, then it cannot be installed on an aircraft that is already present in the set of aircraft with installed engines Li_t, as seen in Equation 42.

$$(a_{e,t-1} \geqslant n+1) \land (p \in Li_t) \implies a_{e,t} \neq p \quad \forall p \in P \tag{42}$$

3.8.11 Engine change constraint

As engines generally are not exchanged among aircraft without a maintenance process in between, it is beneficial for the training process to constrain the possibility of directly changing from one operational action to another. Consequently, engines in operation may stay in operation at the same aircraft, go to a shop visit action or to storage, as presented by Equation 43.

$$(a_{e,t-1} < n+1) \implies a_{t,e} = a_{t-1,e} \text{ or } a_{t,e} \geqslant n+1 \tag{43}$$

4 Methodology

The ESV scheduling problem, being formulated as a MDP, can then be solved using reinforcement learning. The implementation of the reinforcement learning optimization is here presented through the definition of the problem assumptions, training environment, training optimization tools and the creation of initial scenarios for each episode. At last, a greedy policy is defined to be used as base for a comparison to the reinforcement learning results, which is done my evaluation the different defined key performance indicators.

The methodology for the engine shop visit scheduling and optimization process is summarized by the flowchart seen in Figure 5.

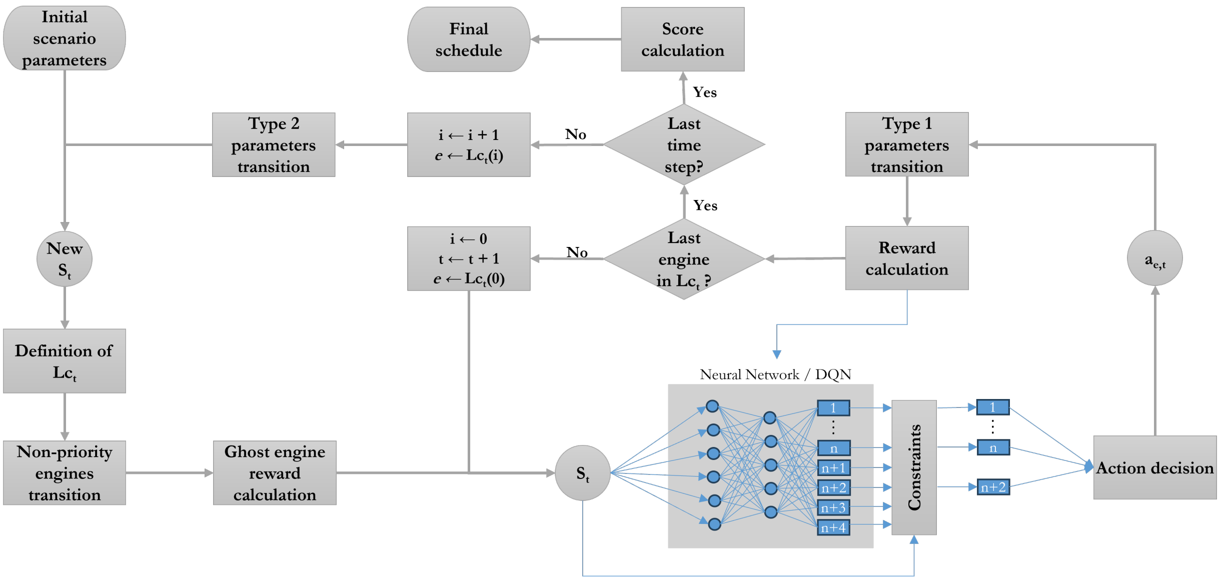


Figure 5: Flowchart for the ESV scheduling optimization process through the proposed reinforcement learning methodology.

4.1 Assumptions

In order to represent the complex real life ESV scheduling process into mathematical models that could be implemented in a virtual environment, a few assumptions were necessary:

- As the thrust setting is one of the factors that affect performance deterioration, each engine will remain at their current thrust setting for the reminder of their life cycle;
- Each aircraft will remain operating at the same type of routes and environment for the remainder of their life cycle;
- As a consequence of the previous assumption, it is also considered that the EGTM deterioration rate and the flight cycles per day average can be defined as constants;
- The shop visit turn around time ϕ is a constant and will be the same independently of the period at which the SV happens, the available capacity at that moment, and the workscope of the maintenance;
- Engine deterioration rate is linear and independent from engine age;
- The operator will never opt to install two engines at one aircraft at the same time;
- Random failures are not considered;
- The wing at which an engine is installed (left or right wing) has no impact on the optimization;
- All aircraft have only two engines spots and need both to be operational;

4.2 Training environment

The problem formulation was implemented in Python as an environment using OpenAIs Gym library. Its structure consists in a main step function, a reset function, functions for the initial scenario definition, the reward function and complementary functions used by the step function to update the state space. A reinforcement learning agent is used to interact with the environment and learn optimal decision-making policies for the ESV scheduling problem.

The agent interacts with this environment by observing the state space, which represent various scenarios in engine scheduling, and taking actions within the possibilities presented by the action space. The consequences of these actions, along with calculated rewards, are fed back to the agent. The reinforcement learning agent learns through this iterative process of action and feedback. The agent's learning process is driven by the objective of finding a policy, a set of rules dictating the best action to take in each state, that maximizes the cumulative reward over time, thus achieving the most efficient and effective scheduling outcomes, based on the episodes final scores.

In this paper, the reinforcement learning approach was implemented with a Deep Q-network (DQN) learning algorithm to define the optimal policy. In it, it uses a deep convolutional artificial neural network to estimate Q-Values for each action, based on the current state space [Sutton and Barto, 2018]. By doing so, it is able to deal with larger and more complex state spaces, as the case for the ESV scheduling problem.

All learning processes and tests were performed using a computer equipped with a Intel Core i5-7300U CPU @ 2.60GHz, 16GB or RAM and 256GB of memory.

4.2.1 Prioritization

When dealing with large fleets, letting the reinforcement learning agent decide each action for each engine would lead to a very slow learning process, as each action has to be passed through the environment and its consequences back-propagated through the agent's neural network. Moreover, as an action has to be decided for every engine at every time step, a higher number of engines and time steps will lead to larger episodes, consequently reducing the number of episodes for the same number of training steps.

As the scheduling problem presents a stable system, where most of the time engines will stay at their action for long periods of time, it is not necessary to use the agent to choose each action for each engine, rather, the agent should be able to choose an action only for those engines that present a higher probability of requiring a change in their current state.

This prioritization process has to use a policy to determine which engines should have their action chosen by the agent. At each time step progression, a new set of "critical engines" Lc_t has to be defined, these engines will in turn have an action decided by the agent. The policy is defined by six different criteria, each criteria has a maximum number of engines that it can include to the set of critical engines. The critical definition algorithm has its general logic presented in the way of pseudo-code in Appendix C. Section 4.2.2 presents the logic for all criteria.

4.2.2 Critical engines definition

Each of the criteria is defined as follows:

- Back from shop visit:

Once an engine is sent to a shop visit, the state space for the following Φ time steps for the corresponding engine are set to reflect that state. As the engine that has been sent to maintenance will necessarily remain in the SV action for the entire turn around period, setting it as a critical engine would not generate any learning benefits. However, once the engine has returned from a shop visit it has no default action, as it could go to storage or be reinstalled in any aircraft. It is thus necessary that all engines returning from a shop visit at the current time step are considered critical engines. The maximum number of engines chosen by this criteria is the same as the MRO capacity $M_{e,t}$, as that is the maximum number of engines that might return from a SV at once.

- Non-operational engines in storage:

Once an engine reaches its degradation status threshold, it has to be removed from operations, as seen in constraint 3.8.2, and put in storage or be sent to a shop visit. In case it is sent to storage, these engines will not be able to return to operations unless they are sent to a shop visit at some point and, thus, it is necessary that they are among the critical engines. The agent is then able to send these engines to a shop visit at the time step it deems optimal. The maximum number of engines chosen by this criteria, once again, is equal to the MRO capacity $M_{e,t}$, as this is the maximum number of engines that might be sent to maintenance at any given time step.

- Lease ending:

Engines close to (or at) their leasing end period are subjected to two different constraints that might require a change on their current action, them being constraint 3.8.8 and 3.8.9. Engines close to the leasing progress moment at which they have to be at their titled aircraft must be included in the critical engine list, so that they have the opportunity to be removed from another aircraft and reinstalled at their titled aircraft if so necessary. Furthermore, engines that reach their leasing end have to be removed from operations and sent to a permanent storage action and, thus, also have to be in the critical engine list. To do so, all engines within six time steps from their titled aircraft leasing point are arranged in order of proximity to the point, and the four closest (if there are more than four) are selected. Similarly, all engines that are within two time steps from their leasing end point are arranged in order of proximity to it. Once again, the four closest (if there are more than four) are selected. Once an engine reaches its leasing end point and is removed from operations, it may not be further considered when forming the critical engines list.

- Best engines in storage

As previously stated, once an aircraft has an engine removed, a spare engine has to be installed on its place in order to keep the aircraft operational. The engines in storage with lowest degradation statuses allow for a higher flexibility in this replacement, possibly staying at the aircraft for a longer period if necessary. It is thus necessary to arrange all engines in storage in order of highest critical status, where the critical status is the lowest value among the LLP and EGTM statuses. From this list, the four best engines are selected as critical engines.

- Worse degradation engines

From all engines, the ones that will most frequently require a change of action in the following time step are the ones in operation with low deterioration statuses, as they will most likely have to be replaced sooner than others and be sent to storage or maintenance. To account for these engines, all engines in operation are arranged in order of lowest critical status. Some critical engine definition criteria set a maximum limit on the number of engines to be included in the critical engine set, without specifying a minimumpotentially resulting in no engines being added. Given that the total number of critical engines is a fixed value, the quantity of engines selected based on the worst degradation criteria is determined by the number needed to fill the remaining slots in the list to reach its predetermined size. The number of engines in the priority list has to be set as a large enough value so that, even if all other criteria send their maximum number of engines to the list, the worse degradation criteria still has at least four engine spots to be added.

4.2.3 Next time step preparation

With prioritization, all engines that are not considered critical must be updated in the state space considering that they will remain at their current action. To do so, at the end of each time step, the new list of critical engines Lc_t is determined and the remaining engines go through an internal step function, updating their actions and statuses in the state space for the following time step in accordance to the transition functions presented in Section 3.6. The critical engines remain untouched and can be identified in the state space, after the internal step for all non-priority engines is executed, by having zeroes at their action indicator.

Considering that most of the episode's score is generated by the normal operation of aircraft, only evaluation critical engines would lead to bad score performance, as these engines will frequently be in actions that generate penalties, as shop visits, or just will not generate any positive reward, as while staying in storage, for example. This situation could hinder the learning process, as the agent would likely fail to recognize the long-term benefits of executing a shop visit. Therefore, this would likely result in the policy avoiding maintenance actions all together.

To account for the non-priority engines reward without having the agent to go through them in the simulation process, a " ghost engine" was implemented in conjunction with a "ghost action". The ghost engine is always set as the first engine in the critical engines list and presents the agent with the updated environment for the new time step, after all non-priority engines have been updated. The ghost action has no effect over the state space and all other actions are masked, so that it is the only possible action for the ghost engine. For all other engines the ghost engine is masked and cannot be applied. The reward for the action of all non-priority engines are summed and sent to the agent as reward of the ghost engine, so that all normal operation engines are accounted for.

4.2.4 Action decision

To restrain the agent to only choose actions that respect the set of constraints seen in 3.8, at every decision step the state space is analyzed and actions that violate the constraints are removed from the list of possible actions. Concurrently, their respective Q-Values are also set to minus infinity.

Actions are decided according to an ϵ-greedy exploration vs. exploitation balancing strategy. In it, the epsilon value is a parameter that defines the likelihood of the agent taking a random action, instead of taking the best-known action, identified in the DQN by having the highest Q-value. The value of ϵ is linearly reduced from a defined initial exploration rate ϵ_0, at the beginning of the training process, to a final exploration rate ϵ_f at the end of the training process.

4.3 Scenario generation

The learning process requires that different scenarios, within the bounds of realistic possibilities, are presented to the agent, so that it is able to observe and learn from patterns in the state space, actions and received rewards. To do so, a random initial scenario function is used to create a different initial condition for each episode during the learning process. The reinforcement learning agent will try to improve the episode's final score, thus going through different initial scenarios allow the agent to avoid overfitting to one particular combination of parameters.

4.3.1 Initial scenario

The initial state space is created starting by the time step immediately previous to the starting point, hereby named time step zero. Inside the engine state, all engines are ordered in sequence of titled aircraft, so that variables z_e are in ascending order, and have their respective action indicator $a_{e,t}$ initially defined as their respective titled aircraft. Spare engines are assigned to the storage action or a shop visit action, that being selected through random chance, as long as there is MRO capacity available. Based on historical data it was observed that, for a usual operations day, between 70% and 95% of all engines are installed at their respective titled aircraft. To simulate that, at every initial scenario generation, the number of engines to have their positions swapped is randomly selected within the boundaries seen in the historical data. Once the number of engines to be swapped is defined, the particular engines that are to be swapped are also randomly selected and their position are changed accordingly. Consequently, at the first time step, all aircraft are in operation with two engines assigned to them, thus the engine counter variable $k_{p,t}$ is set as one for all aircraft.

Regarding the LLP and EGTM statuses, every engine is attributed a random deterioration status between 10% and 95%. Once all engines have a random degradation status defined for both parameters, the previous five time steps are also set, having as base the defined time step zero. All engines in operation and in storage will automatically be assigned to the same action for the previous time steps. For engines that were defined as being in a shop visit at time step zero, it has to be defined for how long it has been in maintenance up to that point, that being decided by randomly selecting a value between one and ϕ. If time step zero was defined as being the first day of a shop visit, then all previous time steps are set as storage actions. Likewise, statuses for LLP and EGTM are defined based on the action taken by each engine for that time step. Engines in operation will restore their degradation by γ and σ, engines in storage will remain at the same degradation and engines in a shop visit will have their degradation statuses calculated based on the amount of time steps in maintenance previously defined and a random degradation status for the time step previous to the start of the shop visit process. Based on the defined deterioration statuses for the previous time steps, the average rates of decay $\overline{\lambda}_{e,t}$ and $\overline{\xi}_{e,t}$ are calculated in accordance to Equations 23 and 24. Similarly, the critical status $CS_{e,t}$ is also calculated based on the time step zero deterioration statuses in accordance to Equation

The AOG indicators $\Theta_{p,t}$ are set to 0, as the engines have not yet have their action assigned to the current first time step.

For the progress of leasing period for each engine, the leasing end point for each engine has to be defined. For training, all engines have the same leasing period Δt_l, however with a variable ending point. Engines with the same titled aircraft are assigned the same leasing ending point. The leasing ending point $t_{l_{end}}$ is defined through a random variable from a range between 8 time steps before the last simulation date and 100 time steps after. The leasing progress can than be then calculated via Equation 44.

$$l_e = \frac{\Delta t_l - t_{l_{end}}}{\Delta t_l} \tag{44}$$

Although the time progression between two time steps can be any period of time, for this paper it is considered one week. At the start of a new scenario, a random value between one and 52 is chosen. Based on this value and an arbitrary assumption that summer starts at week 27 and has a length of 8 weeks, the summer approach variable τ_t and the summer progression variable ρ_t can be calculated.

4.3.2 Basic Scenario

As this type of environment, to the best of the author's knowledge, has not yet been implemented with reinforcement learning before, it was gradually implemented in steps. First a basic scenario was defined, with a reduced number of engines, aircraft, and constraints, in order to test if the method was able to interpret and act upon the suggested structure of state space and action space.

For the basic case, 2 aircraft and 5 engines were used, thus having one spare engine. The state space was reduced, not presenting the titled aircraft, the AOG indicator, the leasing progress, the summer approach and summer progress variables. Only the constraints strictly necessary to create a realistic basic schedule were considered, them being constraints 3.8.2, 3.8.3, 3.8.8 and 3.8.5. All parameters were considered deterministic. The deterioration rate was fixed at 0.01 per day for both LLP and EGTM and the statuses after restoration for both parameters were set to 1. Furthermore, the recovery during shop visit was not gradual, being fully restored in the first shop visit time step and then constant up to the last. It was considered that the MRO provider took three time steps to return the engine and that it was capable of only performing maintenance at one engine at a time. The simulation time was 80 time steps.

The implemented reward function for the basic scenario was a simplification of the reward function presented in Subsection 3.7. In this case, penalties for not using titled engines (Pn_t), summer changes (Pn_{summer}) and for AOG (Pn_{AOG}) were not considered. All three shop visit actions were assigned the same penalty value.

4.4 Greedy Policy

In order to evaluate if the policy reached by the agent is good, a greedy policy was elaborated for comparison. In it, each engine only goes to a shop visit once it reaches its operational limit and will only perform the shop visit correspondent to the parameters responsible for triggering the shop visit. Once the engine comes back from a shop visit, it can only be assigned to its titled aircraft. If that aircraft already has two engines assigned to it, it will be sent to storage, remaining in that action until the aircraft has a spot available. Spare engines, if available, will assume the position of any aircraft without an engine, once all regular engines have their actions assigned.

As this policy only sends engines to maintenance due to their deterioration parameters reaching their operational limits, the realistic action threshold constraint 3.8.1 is not required and, thus, not implemented. Furthermore, due to limitations in the algorithm of the policy, the double installation constraint 3.8.10 was also not implemented. At last, as a consequence of the logic of the greedy policy, engines can only be assigned an operational action if they are in storage or returning from maintenance and, thus, the engine change constraint is spontaneously followed.

The algorithm for the greedy policy can be seen in form of pseudo code in Appendix D.

4.5 Neural Network

The neural network was implemented using the *TensorFlow* platform and the *Keras* API in Python. For all scenarios, two hidden layers were used using the rectified linear unit (ReLU) activation function, while the output layer uses the linear activation function. The state space is flattened into a 1D array before being sent as input to the neural network. Neurons are fully connected between layers. The Deep Q-Network was implemented using the default module from the "keras-rl" package.

The use of a very large state space has direct impact on the reinforcement learning process. By using fully connected layers in the neural network, every new parameter in the state space will generate N_1 new connections to the first layer, and thus also N_1 new weights that have to be considered in the network and have a weight determined through the training process, where N_1 is the number of nodes in the first layer.

4.6 Key Performance Indicators

When considering the scheduling of engine shop visits, the main performance indicator will be the total costs for the entire life cycle of a single engine. When considering multiple engines, with different ages, leasing agreements and performance parameters, calculating and comparing costs might prove to be complex, as well as frequently inaccurate. To help in the comparison of scenarios, other key performance indicators (KPIs) might be beneficial, such as:

- **Average aircraft operational availability:** As aircraft operations is the main source of revenue for any airline, keeping airplanes operational as extensively as feasible is a high priority. At every time step, after all engines have had an action assigned to them, all aircraft with Q engines assigned to them are considered operational. Aircraft operational availability is here defined by the ratio between the number of operational time steps for an aircraft to the total number of days analyzed, being thus a value between zero and one. The average of operational availability for all aircraft in the fleet is then the average aircraft operational availability for the analyzed schedule.

- **Number of spare engines required:** While spare engines provide flexibility for the airline to send engines for a shop visit while maintaining its aircraft operational, their considerable costs make it unfeasible to keep a large amount of them in storage, waiting to be used as replacement for titled engines in maintenance. It is thus ideal to keep the number of spare engines as low as possible while keeping aircraft operational availability as large as possible.

- **Average engine utilization:** Similarly to the case for aircraft, the engine utilization provides the operator with a metric to evaluate how well their engines are being utilized. If in a defined schedule the engine utilization is low, that might indicate that there are too many engines, or that they are not being utilized optimally. To further explore these possibilities, two secondary KPIs are also useful:

  - **Average titled engine utilization:** This KPI provides the operator with a deeper understanding of how the engines are being utilized. Generally, titled engines should have a higher utilization rate than spare engines.

  - **Average spare engine utilization:** A low utilization rate of spare engines might indicate that there are too many spare engines available, and that an optimal solution could be found with less, reducing leasing costs.

- **Number of AOGs:** The frequency of an aircraft's operational downtime is a key metric for operators. It allows them to compare different scheduling solutions and identify which ones minimize operational impact. This metric is essential for balancing maintenance needs with operational efficiency.

5 Results

Three different types of tests were performed to evaluate the reinforcement learning approach to the ESV scheduling problem.

5.1 Base scenario

In order to verify if the proposed environment is able to be interpreted by the reinforcement learning agent and achieve the optimal result for a small case scenario, a base case was created.

5.1.1 Initial conditions

The initial deterioration conditions for the base scenario was determined with random initial values, except for the spare engine, which was manually placed at a pair shop visit, so that its deterioration parameters were maximal at the start of the simulation. The scenario followed the structure presented by Sections 3 and 4, however without prioritization. The parameters for the time step zero are presented in Table 1.

Engine	Action	LLP Status	EGTM Status
1	1	0.63	0.52
2	1	0.88	0.22
3	2	0.38	0.36
4	2	0.63	0.28
5	6	1.00	1.00

Table 1: Base scenario initial parameters

This base scenario was manually optimized by attributing each engine to its previous action, up to the point where a degradation threshold was reached. Once this threshold was achieved, the engine was always assigned to a pair shop visit, as this is the optimal decision when considering that all shop visit actions have the same cost for this environment. Once an engine returned from a shop visit, it was kept in storage as a spare engine. Once another engine reached its threshold and had to be replaced, the engine is storage would then be installed. The last engine to reach its threshold was then kept in storage with its last status, as the spare engine had enough LLP remaining useful life and EGTM to complete the simulation period. This was implemented in a spreadsheet in order to evaluate all constraints in each time step and calculate the total score.

The manual optimization reached a total calculated score of 159.1. This score can be broken down into the maximum possible score achieved by two aircraft in eighty time steps, with one point for each aircraft in operation at each time step, totaling 160, and the maintenance penalty for three different maintenance period, with a TAT of three time steps and a penalty of 0.1 per time step in maintenance, totaling 0.9. When subtracting the minimal penalty (while guaranteeing all aircraft operational for 100% of the simulation period) from the maximal score, the achieved result is obtained.

To test the functionality of the virtually implemented problem formulation, evaluating if the actions provoke the projected consequences, and compare the manual optimization to the reinforcement learning optimization, each action from the defined sequence of actions was manually set at each decision point into the environment. The position of all engines can be seen for each time step through the schedule presented in Figure 6.

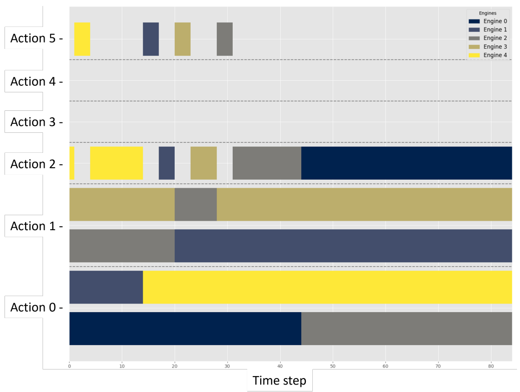
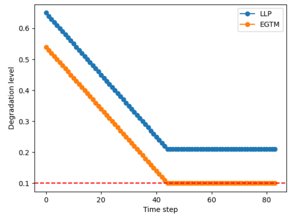


Figure 6: Engine action by time step for manual optimization.

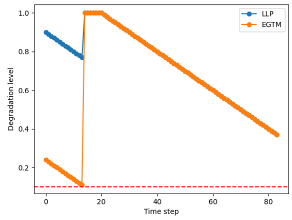
From the schedule it is possible to see that aircraft A and B, respectively represented by action zero and one, have two engines assigned to them at all times. As there are only four operational positions, the remaining engine is always seen in the pair shop visit action (action five) or in the storage action (action two).

The degradation status for each engine is seen in Figures 7a to 7e. The dotted red lines in the deterioration plots represent the maximum degradation status for operational service, which for the base scenario was set at 10%. It is possible to see that the manually obtained solution respects constraint 3.8.2, never operating below the threshold. Furthermore, the pair shop visit action correctly led to a complete restoration of LLP and EGTM statuses. Engines in operational action with two engines assigned to that action had their LLP and EGTM statuses continually degraded through time, at the same pace predicted through the spreadsheet. When in storage, no degradation is visible, as expected.

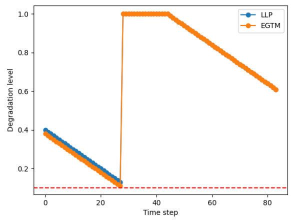
At last, the calculated score for the entire episode also matched the estimated value in the spreadsheet, at 159.1, ratifying the reward calculation function.



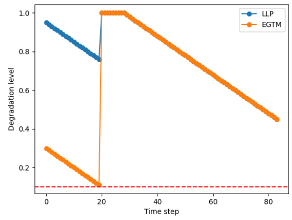
(a) Engine 0 deterioration.



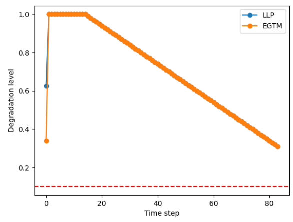
(b) Engine 1 deterioration.



(c) Engine 2 deterioration.



(d) Engine 3 deterioration.



(e) Engine 4 deterioration.

Figure 7: Deterioration statuses per time step for all engines in manual optimization.

5.1.2 Reinforcement Learning Parameters

The parameters to perform the learning process for the Deep Q-Network were defined through testing, running the training process with different parameter values and evaluating their effect on the final results. The final architecture was defined with two hidden layers using rectified linear activation functions. All final learning parameters for the base case scenario are summarized in Table 2.

Parameter	Value
Nodes in 1^{st} layer	220
Nodes in 2^{nd} layer	170
Learning rate	$2.7 \cdot 10^{-6}$
Initial exploration rate ϵ_0	0.45
Final exploration rate ϵ_f	0.02
Target network update frequency	50 steps
Discount factor γ	0.999
Number of steps	200.000
Number of episodes	500

Table 2: Reinforcement learning parameters for base scenario.

5.1.3 Results

After training, the achieved policy was used to try to optimize the base scenario, using the same initial parameters as for the manual optimization. The position of all engines can be seen for each time step through the schedule presented in Figure 8.

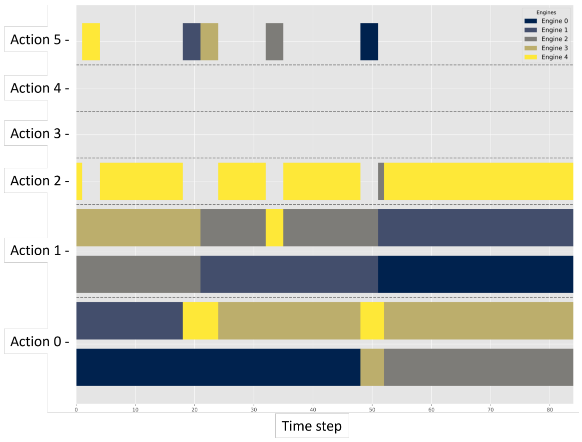
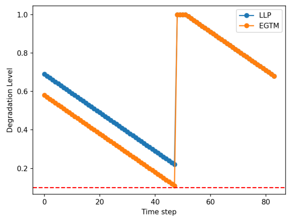
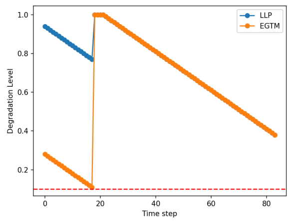


Figure 8: Engine action by time step for reinforcement learning optimization.

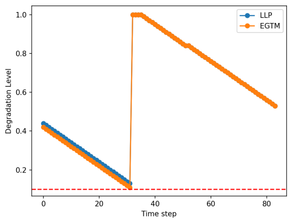
The degradation status for each engine is seen in Figures 9a to 9e.



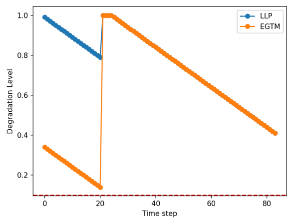
(a) Engine 0 deterioration.



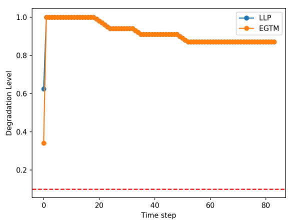
(b) Engine 1 deterioration.



(c) Engine 2 deterioration.



(d) Engine 3 deterioration.



(e) Engine 4 deterioration.

Figure 9: Deterioration statuses per time step for all engines in reinforcement learning optimization.

The total score for the episode was 158.8.

In addition to the training process, a random action decision policy was used to compare the advance on the scores along the training process with the use of random actions. The random policy, however, was only able to randomly pick one of the actions available in the action space after its reduction by the constraints application. This comparison can be visualized in Figure 10.

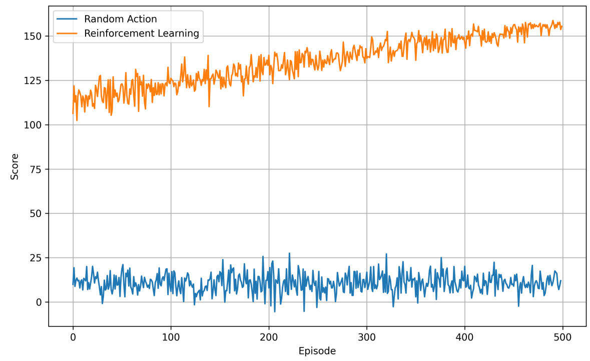


Figure 10: Episodes scores during learning process and for random actions.

5.1.4 Discussion

Regarding the fine tuning of the reinforcement learning parameters, the optimization process demonstrated that a very low learning rate, a high discount factor and a lower than usual initial exploration rate are required. This could be explained from the nature of this environment, where most engines will remain at the same action for long periods of time, and only occasionally will go through different actions. The low initial exploration rate allows for the agent to observe a more stable system, and thus closer to normal operations, for a longer period of time. This increased stability facilitates the understanding of the individual effects of different engine actions to the system, as their effect is correlated. Being a scheduling process with the goal of reducing overall costs for the entire life-cycle of engines, using a higher discount factor γ is essential for the agent to take into consideration the long term effects of current decisions.

Based on the created schedule and the score comparison with random actions, it is noticeable that the agent is able to perceive through learning how the action decision process works, realizing which engine is currently active. It actively tries to keep engines in operation and send them to maintenance when (or before) they reach the defined degradation threshold. Furthermore, it is clear the agent is able to understand the long term benefit of a maintenance procedure, even though it results in a negative reward at the moment of action.

In terms of optimization, it is observable that the solution achieved by the agent is slightly higher than the manual optimization process. Reinforcement learning, however, will not necessarily lead to optimal solutions, being a good and flexible solution enough. The solution, however, is close to the theoretical maximum, using one extra maintenance opportunity. Although there is no separation of normal engines and spare engines in the state space of this basic scenario, the policy created by the reinforcement learning process only uses engine number 4 as a spare engine, quickly returning the other engines into service once they are back from a shop visit. This seems to indicate that, although the policy understand the problem‘s logic, it is not able to also take into consideration the simulation period, and how the policy should be affected by its end approaching. For example, the manual optimization decided not to perform the shop visit on engine 0 because, for the simulation period in evaluation, it would not be necessary.

Although all shop visit options are available, the optimization process is able to come to the same logical decision of only using pair shop visits, as described in the manual optimization, as they have the same cost as the other shop visit processes.

Two optimization actions are noteworthy of attention. First, engine 3 is sent to a shop visit before it reaches its threshold. Although it is impossible to know the the logical process of the trained policy, this action seems to be correlated to the end of the shop visit action of engine number 1, freeing an operational space at aircraft B that is filled with engine 1 and maintaining engine 4 in aircraft A for longer, reducing by one the number of necessary interventions and being thus an example of opportunistic replacement, even though this action, for this environment, has no impact on the total score. Secondly, engine 2 is removed from aircraft B and installed in aircraft A, although this is not necessary in this environment. This might be correlated to the start of the scenario, where engines 0 and 1 and 2 and 3 are together at the same aircraft and, by doing so, they also end at the same aircraft. By using the spare engine, this aircraft change has no effect on the episode's score and is in fact a valuable policy, as for full scenario optimizations this is one of the constraints.

5.2 Full scenario

Once the base scenario test demonstrated the viability of the method, the method was tested with the complete state space and the entire set set of constraints.

5.2.1 Initial conditions

For a full test of the method, the size of the fleet and number of engines was increased from the base scenario to numbers that more closely resemble the values observed in mid-size airlines. The values used for these parameters for the learning and testing processes are presented in Table 3. The episode length and the size of the set of critical engine were decided through trial and error, exploring different values up to the point that a good trade-off between training duration and final results was reached.

Parameter	Value
Episode length [time steps]	80
Turn around time - ϕ	5
MRO maximum capacity - M_{max}	3
Number of aircraft - n	30
Number of regular engines	60
Number of spare engines	6
Number of critical engines	20

Table 3: Training environment scope parameters.

Other parameters have to be decided from operational historic data. From the parameters presented in Table 4, the LLP decay rate γ and the EGTM decay rate σ are approximations based on data from a major European airline. The values for the reward and penalties were once again reached through trial and error, exploring different values that lead to a higher average score and aircraft utilization rates.

Parameter	Value
LLP decay rate - γ	$1.05 \cdot 10^{-3}$
EGTM decay rate - σ	$7 \cdot 10^{-4}$
Titled engines reward - Rv	1
LLP ESV penalty - c_l	0.05
Performance ESV penalty - c_e	0.05
Pair ESV penalty - c_p	0.08
Non-titled engine penalty - Pn_t	0.05
AOG Penalty - c_a	0.2
Summer Penalty - c_s	0.2

Table 4: Operational data parameters.

Furthermore, this initial testing scenario was optimized multiple times in order to fine tune the reinforcement learning parameters. The parameters that had the best performance are presented in Table 5.

Parameter	Value
Nodes in 1^{st} layer	280
Nodes in 2^{nd} layer	220
Learning rate	$2 \cdot 10^{-3}$
Initial exploration rate ϵ_0	0.3
Final exploration rate ϵ_f	0.0001
Target network update frequency	5500 steps
Discount factor	0.9999
Number of steps	400.000
Number of episodes	238

Table 5: Reinforcement learning training parameters.

5.2.2 Results

The training process took 1 hour and 57 minutes. The score of each episode during training can be seen in comparison to random decisions and the greedy policy in Figure 11.

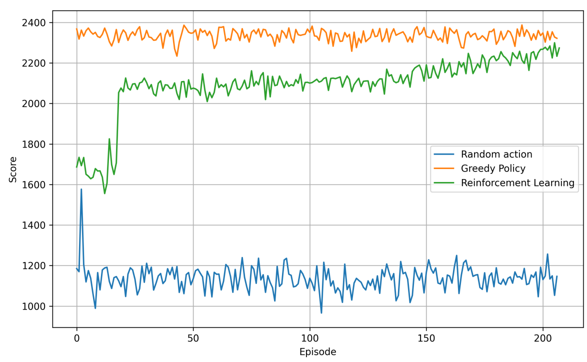


Figure 11: Episodes scores during learning process and for random actions for full scenario.

After training, finding the optimal solution for a random initial scenario took on average 15 seconds. The test for all policies was repeated 100 times and the average results recorded. For the case of the reinforcement learning optimized policy, results are presented in Table 6.

Parameter	Value	Std. Dev.
Avg. score	2303.22	26.22
Avg. acft. utilization	99.45%	0.59%
Avg. eng. utilization	90.64%	0.29%
Avg. reg. eng. util.	94.03%	1.08%
Avg. spare eng. uti.	56.74%	9.02%
Avg. numb. of AOGs	52.41	16.50

Table 6: Average results for 80 steps with RL.

When using the greedy policy, the results for the same number of simulations are presented in Table 7.

Parameter	Value	Std. Dev.
Avg. score	2336.92	28.07
Avg. acft. utilization	99.17%	0.75%
Avg. eng. utilization	90.49%	0.39%
Avg. reg. eng. util.	94.59%	1.71%
Avg. spare eng. util.	49.37%	14.58%
Avg. numb. of AOGs	16.31	6.15

Table 7: Average results for 80 steps with greedy policy.

When considering each time step as a week, optimizing a schedule only for the following 80 weeks does not provide a life-cycle schedule for the engine, as an engine might stay over 15 years in service. It was thus tested what is the effect of testing a longer scenario without retraining the policy. The results for 800 time steps are presented in Table 8. After training, finding the optimal solution for a random scenario took on average 154 seconds.

Parameter	Value	Std. Dev.
Avg. score	22634.38	36.07
Avg. acft. utilization	99.81%	0.13%
Avg. eng. utilization	90.82%	0.07%
Avg. reg. eng. utilization	95.97%	0.54%
Avg. spare eng. utilization	39.30%	5.29%
Avg. numb. of AOGs	495.47	22.06

Table 8: Average results for 800 steps with RL.

In Table 9 the results for 800 time steps using the greedy policy are presented.

Parameter	Value	Std. Dev.
Avg. score	23330.02	303.73
Avg. acft. utilization	98.62%	1.02%
Avg. eng. utilization	90.28%	0.46%
Avg. reg. eng. utilization	91.96%	1.46%
Avg. spare eng. utilization	73.53%	10.83%
Avg. numb. of AOGs	197.69	14.25

Table 9: Average results for 800 steps with greedy policy.

5.2.3 Discussion

By analyzing the progress of the score during training and comparing it to the random decision policy, it is clear that the agent is once again able to understand the dynamics of the environment, including the long term benefits of shop visits, even with the prioritization system implemented. The fluctuations noticeable in the scores are expected, as each episode is performed with different initial conditions that might be more or less favorable to a good final score.

Although it is possible to see a clear increase on the score for the last quarter of the training process, part of it is expected just from the fact that less random actions are being taken through the epsilon-greedy strategy. Increasing the number of episodes did not lead to an increase on the final score average when testing the policy. This can be seen in Figure 12, where the average score and aircraft availability was obtained for policies trained for different amounts of time steps. All results consist of averages for 30 episodes and 800 time steps.

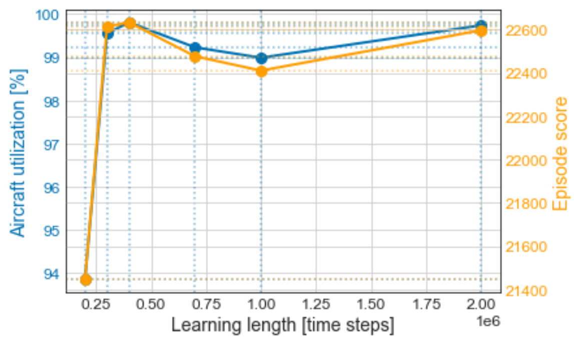


Figure 12: Effect of learning period length on average score and aircraft availability.

Regarding the KPIs, it is clear that the greedy policy provided slightly better scores in both periods simulated. When evaluating the standard deviation, the 800 weeks simulations showed a much higher standard deviation for the greedy policy, showing that it is much more dependent on the initial scenario conditions. The greedy policy, however, provided lower aircraft utilization ratios than the reinforcement learning policy, while both presented low standard deviations. This indicates that the reinforcement learning policy will more effectively utilize engines at airplanes different from the ones they are titled to, what also explains the lower score average, as there is a penalty for using non-titled engines. The very high aircraft utilization rate for the 800 week simulation when using the reinforcement learning policy, in combination to the very low standard deviation, demonstrates that the policy is very flexible to the changes in the randomly generated initial scenarios.

Regarding the engine utilization averages, both policies provided very similar results. The greedy policy had a higher utilization average, what is expected, as the policy has no flexibility to use titled engines on other aircraft. It also had higher standard deviation values, indicating that it is more susceptible to changes in the initial scenario.

It is important to note that the values given to the reward and penalties will have a large impact on the solution. The values used in this simulation were chosen not based on airline data, rather by testing combinations that would improve learning performance. It is thus essential that the airline operator responsible for the schedule optimization also has a high degree of financial awareness, in order to translate the results obtained from the simulation into real revenue and cost estimations.

5.3 Sensitivity Analysis

It was also evaluated how changes in a few of the parameters affect the final results. Two different changes were implemented:

- Change in the number of spare engines;
- Change in the number of aircraft;

The change on the number of engines and aircraft results in the need of the training of a new policy. For all obtained policies, the reinforcement learning learning process maintained the same parameters as the ones presented in Table 5, except for the number of training steps for the change in the number of aircraft, that was increased to 800.000 steps due to the larger state space.

5.3.1 Number of spare engines

Airlines might be able to also save costs by reducing the number of spare engines. It is thus of interest to find the optimal number of spare engines, balancing the gains from the operational flexibility of a higher number of spare engines and their leasing costs. This can be done by evaluating optimal schedules determined with different number of spare engines.

Simulations were performed with two, four, six, and eight spare engines for a fleet of 30 aircraft and 60 titled engines, thus ranging from a margin of 3.26% to 11.76% of spare engines within the entire engine pool. All test were executed testing for 800 time steps and running 100 simulations. Results represent the average for all 100 episodes. The results for two spare engines are presented in Table 10.

Parameter	Value	Std. Dev.
Avg. score	16307.09	861.10
Avg. acft. utilization	71.13%	3.78%
Avg. eng. utilization	81.64%	1.92%
Avg. reg. eng. utilization	81.17%	1.96%
Avg. spare eng. utilization	95.73%	2.29%
Avg. numb. of AOGs	198.42	31.23

Table 10: Results for 2 spare engines

The results for four spare engines are presented in Table 11.

Parameter	Value	Std. Dev.
Avg. score	22578.70	57.90
Avg. acft. utilization	99.54%	0.22%
Avg. eng. utilization	93.52%	0.11%
Avg. reg. eng. utilization	96.28%	0.45%
Avg. spare eng. utilization	52.14%	6.38%
Avg. numb. of AOGs	505.86	22.08

Table 11: Results for 4 spare engines

The results for eight spare engines are presented in Table 12.

Parameter	Value	Std. Dev.
Avg. score	21980.50	104.43
Avg. acft. utilization	96.88%	0.44%
Avg. eng. utilization	86.20%	0.73%
Avg. reg. eng. utilization	94.78%	0.67%
Avg. spare eng. utilization	21.90%	5.47%
Avg. numb. of AOGs	448.29	35.83

Table 12: Results for 8 spare engines

5.3.2 Number of aircraft

To evaluate if the presented method is efficient and flexible enough to be applied for larger fleets, especially in terms of the critical engine definition algorithm, a simulation was performed for 60 aircraft, using 120 titled engines and an additional 12 spare engines, totaling 132 engines. The same prioritization structure as for all other full scenario results, with 20 engines per time step was used. The training process took 12 hours and 25 minutes. All test were executed testing for 800 time steps and running 100 simulations and the results are presented in Table 13.

Parameter	Value	Std. Dev.
Avg. score	45024.74	105.34
Avg. acft. utilization	98.97%	0.21%
Avg. eng. utilization	90.35%	0.15%
Avg. reg. eng. utilization	94.53%	0.50%
Avg. spare eng. utilization	48.54%	4.87%
Avg. numb. of AOGs	783.41	48.57

Table 13: Results for reinforcement learning policy

The same scenario was also used using the greedy policy, producing the results presented in Table 14.

Parameter	Value	Std. Dev.
Avg. score	45370.05	1110.67
Avg. acft. utilization	96.20%	2.15%
Avg. eng. utilization	89.17%	0.98%
Avg. reg. eng. utilization	89.56%	1.30%
Avg. spare eng. utilization	85.28%	5.16%
Avg. numb. of AOGs	375.57	26.61

Table 14: Results for greedy policy

5.3.3 Discussion

Regarding the number of spare engines, it is clearly observable that both the score and the aircraft utilization average increase from two to six spare engines, with a slight decrease when 8 spare engines are available. Furthermore, results for six spare engines presented the lowest standard deviation, indicating that it provides the highest degree of flexibility to any operational disruptions to the policy. The results for four engines, however, also presented high average score and low standard deviation, indicating that, depending on engine leasing costs, the optimal economic efficiency will likely be found between four and six engines. Noticeable, the average utilization of spare engines with four spare engines was still around 50%, indicating that for about half of the evaluated period they were not used.

As for the number of aircraft, the increase in number of engines and aircraft led to a drastic increase in the training time, a consequence of the significantly larger input state space and the increased length of training. The results showed a better performance of the reinforcement learning than the greedy policy in most of the parameters, except in the average score and number of AOGs. The test proved that the critical engine definition algorithm is still able to efficiently handle the set of engines, allowing the reinforcement learning to effectively maintain aircraft in operation for most of the time.

6 Conclusions

In this paper, the engine shop visit scheduling process was explored, investigating the possibility of using a reinforcement learning approach to help in the optimization process of the scheduling of this dynamic and complex problem. The study demonstrated the viability and effectiveness of the approach, as well as its limitation, being a potentially beneficial tool for airlines to save on maintenance cost.

The application of the proposed algorithm resulted in high aircraft utilization averages, one of the main goal for any airline. The different tests showed the flexibility of the presented method, being able to actively react to the different initial scenarios generated. Notably, the reinforcement learning approach was able to recognize the long-term implications of maintenance decisions, even when the direct reward of the operation of the engine after the shop visit was not explicitly visible to the learning agent, as is the case with the use of prioritization.

The reinforcement learning approach also presented clear limitations. For example, it performed only marginally better than the greedy policy. Admittedly, a better evaluation would be achieved by comparing the reinforcement learning policy to a real life schedule from the past, providing the reinforcement learning with the initial scenario obtained from historical data for the start of the analysis period, but this analysis was not possible with the provided data. Future research is also necessary to better correlate the score with real expected savings generated by the schedule. Moreover, results were highly dependent on the training parameters, and small changes in the initial environment and reward functions might require a new testing process for the optimal training parameters. Comparing the performance of different training parameters also proved difficult, as the training process is influenced by the type of initial scenario that it faces during its training, having as consequence that two training sessions with the exact same training parameters will lead to different results. For future research, the creation of a batch of different scenarios that are always used in the same order could be beneficial, standardizing the training process and facilitating the evaluation of parameters effects.

In conclusion, the proposed methodology provides a new approach to the engine maintenance scheduling process. The capabilities of reinforcement learning allow for airlines to reach new schedules in seconds if unforeseen events come to light, as long as the number of engines and aircraft remain constant. This tool thus provides airlines with high flexibility, allowing operators to test a multitude of different scenarios and help in decision making processes, as for example how many spare engines to keep. The method may thus potentially help airlines to enhance operational efficiency and cost savings.

References

[Ackert, 2015] Ackert, S. (2015). Keeping score: Analysis of an engines shop visit rate.

[Almgren et al., 2012] Almgren, T., Andréasson, N., Patriksson, M., Strömberg, A.-B., Wojciechowski, A., and Önnheim, M. (2012). The opportunistic replacement problem: theoretical analyses and numerical tests. *Mathematical Methods of Operations Research*, 76(3):289319.

[Andrade et al., 2021] Andrade, P., Silva, C., Ribeiro, B., and Santos, B. F. (2021). Aircraft maintenance check scheduling using reinforcement learning. *Aerospace*, 8(4).

[Boere, 1977] Boere, N. J. (1977). Air canada saves with aircraft maintenance scheduling. *Interfaces*, 7(3):1–13.

[Cai et al., 2016] Cai, J., Li, X., and Chen, X. (2016). Optimization of Aeroengine Shop Visit Decisions Based on Remaining Useful Life and Stochastic Repair Time. *Mathematical Problems in Engineering*, 2016:1–11.

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

[Chen and Sun, 2018] Chen, D. and Sun, J. (2018). Fuel and emission reduction assessment for civil aircraft engine fleet on-wing washing. *Transportation Research Part D: Transport and Environment*, 65:324–331.

[Deng et al., 2020] Deng, Q., Santos, B. F., and Curran, R. (2020). A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281(2):256–273.

[Giorgi et al., 2018] Giorgi, M. G. D., Campilongo, S., and Ficarella, A. (2018). A diagnostics tool for aero-engines health monitoring using machine learning technique. *Energy Procedia*, 148:860–867. ATI 2018 - 73rd Conference of the Italian Thermal Machines Engineering Association.

[Hsu et al., 2010] Hsu, C.-L., Wan, T., and Chen, S.-H. (2010). The study of optimum engine shop visit scheduling via genetic algorithm method. 42:141–148.

[IATA, 2022] IATA (2022). *Airline Maintenance Cost Executive Commentary FY2021.*

[Jorgenson and Radner, 1960] Jorgenson, D. W. and Radner, R. (1960). *Optimal Replacement and Inspection of Stochastically Failing Equipment.* RAND Corporation, Santa Monica, CA.

[Kinnison and Siddiqui, 2013] Kinnison, H. A. and Siddiqui, T. (2013). *Aviation maintenance management.* Mcgraw-Hill, New York.

[Kumar et al., 2000] Kumar, U. D., Crocker, J., Knezevic, J., El-Haram, M., and Service, S. O. (2000). *Reliability, Maintenance and Logistic Support : - A Life Cycle Approach.* Springer Us, New York, Ny.

[Lagos et al., 2020] Lagos, C., Delgado, F., and Klapp, M. A. (2020). Dynamic optimization for airline maintenance operations. *Transportation Science*, 54(4):998–1015.

[Li et al., 2019] Li, Z., Zhong, S., and Lin, L. (2019). An aero-engine life-cycle maintenance policy optimization algorithm: Reinforcement learning approach. *Chinese Journal of Aeronautics*, 32(9):2133–2150.

[Mayordomo et al., 2010] Mayordomo, A. F., Ghobbar, A., Ghijs, S., and Cator, E. (2010). *Engine Management: A Decision Support Tool for Strategic Engine Maintenance Planning.*

[Ryder et al., 2023] Ryder, C. L., Bézier, C., Dacre, H. F., Clarkson, R., Amiridis, V., Marinou, E., Proestakis, E., Kipling, Z., Benedetti, A., Parrington, M., Rémy, S., and Vaughan, M. (2023). Aircraft engine dust ingestion at global airports. *EGUsphere*, 2023:1–34.

[Sun et al., 2023] Sun, J., Sun, Y., Chen, D., and Liu, H. (2023). An integrated life-cycle maintenance modeling and simulation framework for aero engine fleet management decision support. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 237(3):604–616.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* The MIT Press, second edition.

[van der Weide et al., 2022] van der Weide, T., Deng, Q., and Santos, B. F. (2022). Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers Operations Research*, 141:105667.

Appendices

A Appendix 1 - Engine Shop Visit Scheduling Parameters

Table 15: Engine Shop Visit Scheduling Parameters

Parameters	Description
$a_{e,t}$	Action assigned to engine e at time step t
α_p	Aircraft affected by current action
c_a	Relative cost variable for AOG
c_l	Relative cost variable for LLP shop visit
c_e	Relative cost variable for performance shop visit
c_p	Relative cost variable for pair shop visit
c_s	Relative extra cost variable for summer AOG
CSp_e	Critical status prediction for engine e
γ	LLP status decay between two time steps
$k_{p,t}$	Counter of engines for aircraft p at time step t
l_e	Leasing progress status for engine e
$LS_{e,t}$	LLP Status - Ratio of RUL to complete life for engine e at time step t
m	Number of engines in airline's pool of engines
$M_{e,t}$	MRO provider available capacity
n	Number of aircraft in airline's fleet
$\overline{\lambda}_{e,t}$	Average LLP status decay
$\overline{\xi}_{e,t}$	Average EGTM status decay
ϕ	MRO provider turn around time
Pn_{AOG}	Penalty for aircraft on ground
Pn_{sv}	Penalty for shop visit action
Pn_t	Penalty for non titled engines in operation
$PS_{e,t}$	Performance Status - Ratio for current to maximum EGTM for engine e at time step t
r_{l_e}	LLP recovery rate for engine e
r_{p_e}	EGTM recovery rate for engine e
ρ_t	Summer progress indicator at time step t
Rv	Reward for aircraft operation
$Rw_{t,e}$	Total reward for engine e at time step t
σ	EGTM status decay between two time steps
t	Time step
τ_t	Summer approach indicator at time step t
TS	Total score for episode
$\Theta_{p,t}$	Aircraft on ground (AOG) indicator for aircraft p at time step t
z_e	Titled aircraft for engine e

B Appendix 2 - Engine Shop Visit Scheduling Sets

Table 16: Engine Shop Visit Scheduling Sets

Set	Description
$A = \{a \in \mathbb{N} \mid 1 \leqslant a \leqslant n+4\}$	Set of all possible different actions
$O = \{a \in \mathbb{N} \mid 1 \leqslant a < n+1\} \subset A$	Set of all operational actions
$F = \{a \in \mathbb{N} \mid a = n+1\} \subset A$	Set of all shop visit actions
$S = \{a \in \mathbb{N} \mid n+1 < a \leqslant n+4\} \subset A$	Set of all storage actions
$P = \{p \in \mathbb{N} \mid 1 \leqslant p \leqslant n\}$	Set of different aircraft in operators fleet
$E = \{e \in \mathbb{N} \mid 1 \leqslant e \leqslant m\}$	Set of different engines in operators fleet
$Lc_t = \{\}$	Set of critical engines in prioritization at time step t
$Li_t = \{\}$	Set of aircraft with installed engines at time step t
$Lr_t = \{\}$	Set of aircraft with engine removals at time step t

C Appendix 3 - Priority Engine Decision Algorithm

Algorithm 1 Greedy Policy Action Decision

```
E^d ← Set E sorted from highest to lowest critical deterioration
E^u ← Set E sorted from lowest to highest critical deterioration
A ← 4
for Engine e in set of engines E do
    B ← min[4, MROavailablecapacity]
    if Leasing progress < 1 and no turn around time constraint then
        if Previous action was a shop visit then
            Lc_t ← e
        else if Previous action was storage and minimum deterioration threshold = True then
            if Counter_1 ≤ B then
                Lc_t ← e
                Counter_1 ← Counter_1 + 1
            end if
        else if Leasing progress close to titled point l_limit then
            if Counter_2 ≤ A then
                Lc_t ← e
                Counter_2 ← Counter_2 + 1
            end if
        else if Leasing progress close to 1 then
            if Counter_3 ≤ A then
                Lc_t ← e
                Counter_3 ← Counter_3 + 1
            end if
        end if
    end if
end for
for Engine e in set of engines E^d do
    if Previous action was storage and e not in Lc_t then
        if Counter_4 ≤ B then
            Lc_t ← e
            Counter_4 ← Counter_4 + 1
        end if
    end if
end for
for Engine e in set of engines E^u do
    if Engine e not in Lc_t and |Lc_t| < 20 then
        Lc_t ← e
    end if
end for
```

D Appendix 4 - Greedy Policy Action Decision

Algorithm 2 Greedy Policy Action Decision

```
if Current engine = reward engine then
    Action ← Rewardaction
else
    if Degradation status above threshold then
        if Prev. action is ops. action then
            if Prev. aircraft has spot avlb. then
                Action ← Previous_action
            else
                Action ← Storage_action
            end if
        else
            if Eng. is not spare eng. then
                if Titled aircraft has spot avlb. then
                    Action ← Aircraft_action
                else
                    Action ← Storage_action
                end if
            else
                if Exists acft. with spot avlb. then
                    Action ← Aircraft_action
                else
                    Action ← Storage_action
                end if
            end if
        end if
    else
        if MRO has capacity avlb. then
            if LLP Status trigger then
                Action ← LLP SV
            else
                Action ← PERFO. SV
            end if
        else
            Action ← Storage_action
        end if
    end if
end if
```

II

Literature Study
previously graded under AE4020

1

Introduction

This chapter introduces the engine shop visit scheduling problem, first presenting the reasons for its importance, then presenting the purpose of its optimization and this report, followed by the scope of the research, the research framework and the presentation of the structure of the entire report.

1.1. The importance of aircraft engine maintenance.

Aviation has continually become more popular and, at the same time, safer, with a constantly increasing number of flights per year and a constant reduction in fatal accidents since the beginning of the 90's [5]. There are many reasons for this improvement in safety. As the technology developed, so did the engineering of all integrated systems in an aircraft, such as avionics, navigation systems, structures and engines. Developments in research and science also allowed for a deeper knowledge into how human factors impact aviation safety, creating strategies to avoid the main determined factors [52]. One of the majors contributors to this increase in safety was the study of past accidents, allowing for the understanding of what are the main accident inducers and triggers [49]. This deeper understanding of the aircraft systems and main accident sources induced, among other things, the development of better maintenance techniques, passing from a system where maintenance was performed as problems appeared, in the beginning of aviation history, to a current proactive scheduled maintenance program, actively searching for possible points of failure and working on them before an incident or accidents occurs [38]. Among these improvements in aircraft maintenance is the improvement in engine maintenance.

Aircraft engines development was also part on this significant improvement in safety and reliability. One of the main metrics showing that is the IFSD (In Flight Shutdown rate), with a significant reduction among the last decades, as seen in Fig. 1.1 and Fig. 1.2. Just as in the case of the aircraft as a whole, the aircraft engine was also benefited by the improvements in technology, with new materials being implemented and better manufacturing and maintenance techniques. This increased reliability also had a direct impact on aircraft design. As engines in the past were remarkably unreliable, aircraft were designed with a higher amount of engines, so that in case of one or more of them were to fail, the aircraft would still be able to sustain flight [26]. With the increased reliability of jet engines and the eventual implementation of ETOPS (Extended-range Twin-engine Operational Performance Standards), twin-jets like the Boeing 767 and 777 and the Airbus A330 and A350, operating with larger and more complex jet engines, slowly replaced most of the three and four engine jets.

To keep these engines performing in the required levels of safety and reliability, a series of engines maintenance procedures are set in place. According to [35], currently around 11% of an airline's operational costs are related to MRO (Maintenance, Repair and Overhaul), with around 43% of that attributed to engine maintenance costs in 2021, with projections indicating an increase to 48% by 2030. Consequently, it is clear that optimizing engine maintenance procedures may have a significant impact in an airline's operational costs. Among the engine maintenance procedures is the Engine Shop Visit (ESV), defined by Honeywell as "An engine removal, regardless of failure responsibility or maintenance category (scheduled or unscheduled), is classified as a shop visit whenever the subsequent engine maintenance performed prior to re-installation entails: a.) separation of pairs of major mating flanges, or b.) removal of a disk hub or spool." [32]

An ESV will usually be performed because of one of the following reasons [3]:

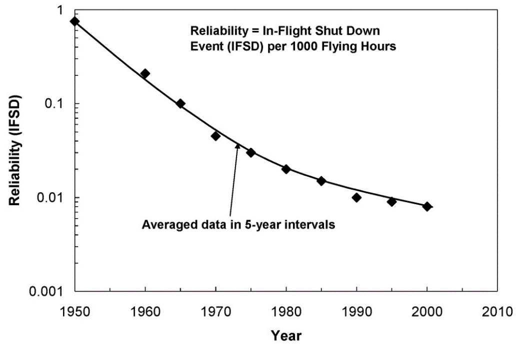


Figure 1.1: In-flight shut down rate from 1950 to 2000 [10]

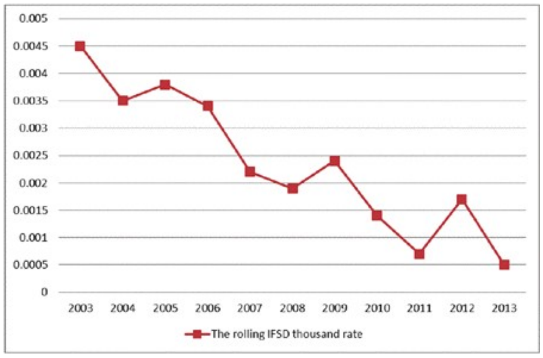


Figure 1.2: In-flight shut down rate from 2003 to 2013 [44]

1. Expiration of a Life-Limited Part (LLP);
2. Performance deterioration;
3. Service bulletin compliance;
4. Emergency Airworthiness Directives;
5. Unexpected engine anomaly.

Items 1-3 can be fulfilled through scheduled interventions. For item 1, the life-limit of the parts are usually set as flight cycles, thus it is possible to estimate when the aircraft will achieve the maximum number of cycles based on its historical flight schedule. Similarly, performance degradation is noted mostly in engine exhaust temperature (EGT), which through different methods to be discussed later can have the time to its margin limit estimated. At last, service bulletins will also generally present a maximum date for the required action to be executed. All these items can be evaluated together in order to schedule the ESV. On the other hand, emergency airworthiness directives and unexpected engine anomalies cannot be predicted, and thus require reactive measures.

1.2. The purpose of ESV scheduling optimization.

An ESV requires a prepared space, high amount of tools, spare parts, human hours and a spare engine. As new generation engines increase in power and complexity, they also increase in price. The scheduling of engine shop visits thus searches for the mutual accomplishment of many factors, trying to use the engines to the maximum extend as possible and, at the same time, avoiding aircraft grounding due to reaching an engine hard limit and reducing the number of spare engines to a minimum, as they are costly. Achieving the ideal balance between all these factors is the base for the ESV scheduling optimization. For small fleets, this type of analysis can be performed manually, but as the scope grows, so does the complexity of the optimization problem, thus finding and evaluating the techniques and methods available to do so is the main purpose of this literature review.

This literature review has the intention of presenting the most important research published in regards to ESV scheduling optimization, or any other scheduling optimization problem that might be of use to perform it. The report aims to present the necessary base to model the scheduling problem and the state of art in modeling and solving of this type of optimization problems.

1.3. Scope of the research

For the purpose of this research, engines are considered in a generalized manner, without any defining characteristics. Further more, it is focused on long-term planning maintenance process, thus considering only heavy maintenance procedures, according to the definition of ESV previously presented.

It is assumed that, at this point, there are no operational and financial limitation from an airline seeking for ESV scheduling optimization and, thus, all literature related to it is considered, independently of method and implementation requirements.

From the 5 ESV reasons previously presented, only LLPs, Performance deterioration and unexpected engine will be considered and further explored, as service bulletins and emergency airworthiness directives can be modeled as scheduled and unscheduled maintenance processes.

1.4. Research framework

The first step into defining the research framework is defining the research objective. The research objective is:

- *Based on engine replacement criteria and the airline's operational and maintenance capabilities, develop a methodology to optimize the scheduling of engine shop visits to reduce the airline's maintenance costs, fulfilling all safety criteria and reducing airline operations disruptions to a minimum.*

As a second step the main research question that has to be answered in order to achieve the research objective is determined. The main research question was defined as:

- *How can maintenance costs be reduced through the optimization of the engine shop visit scheduling for a commercial airline?*

Along with the main research question comes sub-questions that help to structure the research path to answer the main question. Some sub-questions also present sub-questions of their own.

- What factors influence the scheduling of ESVs?
- What are the most used methods to model ESV scheduling?
  - What methods are used for the optimization of ESV scheduling?
- What are the most used methods to model other maintenance processes scheduling?
  - What methods are used for the optimization of other maintenance scheduling problems?
- What factor influence the cost of an ESV?
- What metrics allow for the comparison between methods?

1.5. Structure of the literature review.

This report is divided into 6 chapters.

Chapter 1 presents the introduction to this literature review, presenting the justification for the research, as well as its purpose, its scope and framework.

Chapter 2 focus on presenting the main factors involved with ESV scheduling, starting by presenting the aircraft maintenance process and then proceeding to the specific case of engines maintenance, with the goal of answering the first research question: ”What factors influence the scheduling of ESVs?".

Chapter 3 presents the most used and state of art methods found on literature for the ESV scheduling, starting by presenting the maintenance policies, followed by the presentation of a historical base with early scheduling methods and then presenting research on ESV scheduling optimization.

Chapter 4 presents the most used and state of art methods found on literature for maintenance scheduling and planning for other activities. It is complemented by a discussion on the presented literature and how it might be applicable to development of a new method for ESV scheduling optimization, searching for a research gap on maintenance schedule modeling.

Chapter 5 presents the main optimization methods used to solve the presented different types of maintenance scheduling models. The optimization methods are discussed and a reflection upon which type of methods seem to fit the discussed research gap is performed.

At last, Chapter 6 presents the literature review conclusion, summarizing the work done and the found research gap.

2

ESV scheduling factors

This chapter describes the main factors found in literature that are involved on the scheduling of ESV, in order to answer the second research sub-question " What factors influence the scheduling of ESVs?".

2.1. Aircraft maintenance processes

Although aircraft are in essence a machine with a high number of integrated subsystems, just as many other types of equipment throughout different sectors and industries, the main difference regarding maintenance of aircraft is the high standardization of its processes in comparison to general machinery maintenance [1]. This high standardization was an evolution process of two basic maintenance approaches [38]:

1. The process-oriented approach;
2. The task-oriented approach.

In the beginning of aviation history, maintenance was based in a direct correlation between risk of failure and time in service, limiting the life time of certain parts and, once the limit is reached, removing the piece and overhauling or discarding it, in a concept named as "Hard Time" process (HT). Some less critical parts would be periodically inspected in search of notable degradation, being then also replaced or overhauled, in a concept named as "On Condition" process (OC). These two processes, in conjunction with the "Condition monitoring" (CM) process, where the item cannot have a defined life time or have its degradation level measured or inspected and has to be replaced only once it fails, are the base of the process-oriented approach [60].

The advances in statistical analysis of parts reliability led to more structured maintenance programs, starting with the Maintenance Steering Group (MSG), developed initially by Boeing to be implemented in the maintenance of the then new Boeing 747. Developed together with the suppliers, airlines and the FAA, the MSG was initially based on the classic process-oriented approach and divided the process into six working groups, aggregating different aircraft subsystems. Analyzing each part individually, the MSG system would use a process flow diagram to evaluate if the set part required a HT, OC or CM maintenance process [38]. The MSG system was then generalized and expanded for all aircraft by the Air Transport Association of America, becoming MSG-2, where each aircraft had a list of items to be evaluated, called as "Maintenance Significant Items" (MSIs).

In 1980 MSG-3 was implemented, switching from the "bottom-up" process-oriented approach of the MSG-1/2 to a "top-down" task-oriented approach. This was an evolution from MSG-2, focusing on how failures affect the system as a whole. Each failure is investigated through a different flow chart, evaluating how it affects the operations of the aircraft and if it is a safety concern or has only a financial impact, further down evaluating if the failure is evident or hidden to the crew. This methodology was called "Failure Mode Effect Analysis (FMEA) [12]. Tasks are divided into four sections:

1. Systems and Powerplant;
2. Aircraft Structures;

3. Zonal Inspections;

4. Lighting/High Intensity Radiated Field.

Using the MSG-3 philosophy, type certificate holders (TCH), as aircraft manufacturers for example, will develop together with future operators and the certification authorities the Maintenance Review Board Report (MRBR). The MSG-3 analysis is also implemented to perform the System Safety Analysis (SSA), from which the Certification Maintenance Requirements (CMR) and the Airworthiness Limitations (ALS) documents are elaborated. The MRBR, the CMR and the ALS are the primary sources for the creation of the maintenance program document (MPD), which will be submitted to the aviation agency responsible to certificate the aircraft. These documents have their description summarized below.

- Maintenance Review Board Report (MRBR): According to the FAA [4], the MBRB is “a baseline or framework around which each operator can develop its own individual aircraft maintenance program, (...), outlining the minimum scheduled tasking/interval requirements to be used in the development of an airworthiness maintenance/inspection program for the airframe, engines, systems and components";

- Certification Maintenance Requirements (CMR): According to EASA [27], CMR are defined as “a required periodic task, established during the design certification of the aeroplane as an operating limitation of the type certificate, (...), intended to detect safety-significant latent failures which would, in combination with one or more other specific failures or events, result in a Hazardous or Catastrophic Failure Condition.";

- Airworthiness Limitations (ALS): According to EASA [27], the ALS must contain “Each mandatory modification time, replacement time, structural inspection interval, and related structural inspection procedure."

In addition to the primary sources, secondary sources are also used to elaborate the MPD, as for example Airworthiness Directives, operational requirements, low utilization maintenance program, and others. As a result, the MPD will have all the recurring maintenance information to base the operator‘s Aircraft Maintenance Program (AMP), which will be the base for the scheduling and procedures of all the operator‘s maintenance tasks. The MPD, together with the Aircraft Maintenance Manual (AMM), the operator‘s experience and other non-recurring tasks, are combined to produce the operator‘s AMP. The entire process can be visualized in Fig. 2.1.

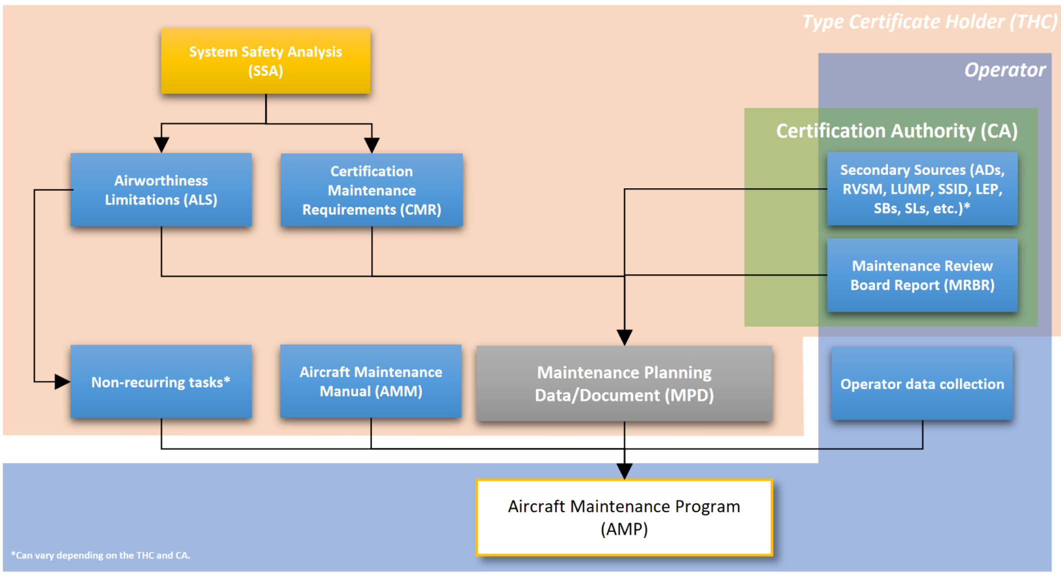


Figure 2.1: Aircraft Maintenance Program construction flowchart.

2.2. Engine maintenance process

As part of the AMP, each scheduled or unscheduled task to be performed on an engine is defines as a simpler maintenance procedure, that can be performed with the engine still installed on the wing, or as a heavier maintenance requiring the removal of the engine is necessary, possibly to perform a shop visit [60]. As the scope of this study is limited to engine shop visits, the processes related to light on-wing maintenance will not be further discussed.

2.3. Engine shop triggers

Understanding the factor that trigger an engine shop visit is essential for the prognostics of when an ESV should be scheduled.

2.3.1. Life-Limited Parts

The Engine life-limited parts are defined by the Code of Federal Regulations, on Section 33.70 of Title 14 [9], as:

> “Engine life-limited parts are rotor and major static structural parts whose primary failure is likely to result in a hazardous engine effect. Typically, engine life-limited parts include, but are not limited to disks, spacers, hubs, shafts, high-pressure casings, and non-redundant mount components.”

According to [48], the number of economically usable flight hours remaining on an LLP at the time it is replaced is called ”stub life". One of the goals of the ESV scheduling optimization is reduce the stub life while complying with maximum phase-out conditions. Furthermore, airlines might have contractual leasing conditions on stub life on an engine at the point of aircraft return, resulting in penalties (mainly financial costs) when they are overrun.

2.3.2. Performance factors

For an aero-engine, the most commonly adopted performance assessment parameter is the Exhaust Gas Temperature Margin (EGTM) [19]. The EGTM is the temperature difference between the measured engine exhaust gas temperature and the red-line temperature, that is, the maximum temperature that the engine should operationally maintain. As an engine continually degrades, the margin reduces as exhaust temperatures go up. This pattern allows for airlines to estimate a time until the engine reaches its limitation and is able to schedule recovery maintenance accordingly [43].

2.3.3. Unexpected engine anomalies

Unexpected engines anomalies are, as the name suggests, causes for the engine to fail, or decrease in performance in such a manner that an engine shop visit is necessary. This type of failure mostly occurs in parts which are under high operational stress, such as compressor and turbine blades and bearings, or caused by foreign object damage (FOD), such as aircraft parts on a termac or bird strikes [[54],[53]].

2.3.4. Maintenance resources

According to [14], one of the main constraints for aircraft maintenance scheduling is the resources available. Many items are among the necessary resources, such as manpower, equipment, hangar capacity and MRO (Maintenance, Repair, and Operations) provider capacity. Another important factor to be taken into consideration is the MRO pace of processing and redelivering the overhauled engines, that is, the shop repair turn-around-time (TAT), as this will determine when the engine is once again available to be put in operation, freeing spare engines to be used in other aircraft [54].

2.3.5. Airline schedule

When evaluating ESV scheduling, being a long term planning, the access to aircraft routing is still not possible as it is not know, being defined only a few weeks before the operation [25]. This makes favoring certain locations more complicated, and therefore the geographical location of the hangar or maintenance base is mostly not taken into consideration, as it is not possible to know where the aircraft will be.

This is not to say, however, that the airline planning has no impact on ESV scheduling, as other temporal and geographic parameters have know and consistent impacts. As noted by [14], it is usual for airlines not to perform maintenance during the summer period, for example, as it is a high demand time, requiring most of the aircraft to be operational, while also being a common holiday season for maintenance staff. Furthermore, aircraft frequently are assigned to a base that does not change frequently, allowing for airlines to predict which maintenance base will be closer to the aircraft at the time on which it is scheduled. This does not imply, however, that other maintenance bases cannot be used, but requires the consideration of the impact of a further maintenance base to the aircraft operation.

3
Main methods of ESV scheduling

This chapter describes the main methods of ESV scheduling found in literature, in order to answer the second research sub-question “What are the most used methods to model ESV scheduling?" and the sub-sub-question ”What methods are used for the optimization of ESV scheduling?".

3.1. Maintenance policies

To understand how a maintenance task is scheduled it is necessary to first understand the difference policies of maintenance that can be performed on a system. [41] separates the policies into 3:

- Failure-Based Maintenance (FBM);
- Time-Based Maintenance (TBM);
- Condition-Based Maintenance (CBM).

3.1.1. Failure-Based Maintenance (FBM)

The FBM is a policy where a maintenance task is only performed when a system part is identified as faulty. The piece is then replaced or restored to the default condition. It is also called Corrective Maintenance (CM) and it is, thus, reactive and unscheduled. This policy has in its advantage the fact that items will be used to the full extend of their operating life, reducing waste of life use (or stub life) to null. As disadvantages, FBMs leave no space to maintenance planning, leaving the operator to perform maintenance at times that might be inconvenient and that might over-stress the maintenance capacity. It might also cause a higher use of resources and generate costly consequences to the system where it is integrated, where it might cause the failure of other items and/or generate longer non-operating periods, with possible loss of revenue and payment of compensation fees due to cancellations.

3.1.2. Time-Based Maintenance (TBM)

On a TBM policy, maintenance tasks are performed at fixed intervals, being also called Preventive Maintenance (PM). These intervals can be determined based on the expected life-cycle of the part and by the operator’s experience. However, once the period is defined, its maintenance will always be performed once the period is over. As advantages, TBM tasks can be planned to an operation optimality, costs of production stop are reduced and safety is improved, mainly on systems where the failure’s consequence is catastrophic. As disadvantages, TMBs might lead to an increase on unnecessary tasks and a consequential increase on costs with labor and items, as the condition of the item is not taken into consideration. It can also increase non-operational times on a life-time analysis when compared to CBM, as well as it does not address failures that are not related to utilization times and might increase the probability of maintenance-induced failures.

3.1.3. Condition-Based Maintenance (CBM)

CBMs, on the other hand, is a newer technique, which tries to overcome the disadvantages found on both FBM and TBM. It is based on prognostics, that is, based on current observations on the system, trying to estimate the Remaining Useful Life (RUL) or the failure probability at a certain point of a certain item. It reduces stub life at the same time that avoids unexpected failures, it requires, however, that the item can be inspected, examined and/or monitored. This continuous evaluation of the item, in general, also incurs in extra costs, being only worth on high cost items or a potential catastrophic failure causer.

3.2. Early maintenance scheduling optimization methods

In the past, maintenance tasks were performed in a on-necessity basis, responding to legislation, operational experience and internal standards [18]. To this date, many airlines still schedule their maintenance manually, using personal experience and spreadsheets, focusing on obtaining a feasible solution and not on generating a solution with a cost-efficiency optimality [42]. This begun to change in the 1960s, with the development of optimization mathematical models that aimed to balance the costs of the maintenance process and the benefits that it generates [18], mainly focusing on searching for optimal preventive maintenance periods and comparing it to corrective maintenance methods.

Early generalist studies, as the one presented by [11], try to evaluate how different maintenance policies compare to each other when dealing with both simple one-element systems and more complex multi-element systems. Using the relation between the required time to perform a scheduled maintenance task and the time required to perform an emergency maintenance task, as well as how frequently emergency maintenance happens, the limiting efficiency of the system can be obtain, that is, what is the portion of time that the system will be operational in comparison to the total evaluation time. For different correlations of maintenance times and failure frequencies, using a simpler method of performing FBM or TMB or using a method of performing minimal repair procedures when a failure occurs and keep operations until a defined elapsed time between maintenance tasks is reached may present better limiting efficiencies for the system. The system can thus be mathematically defined and an optimal method and maintenance interval can be obtain by solving the modeled equations.

Complex systems such as an aircraft, however, are usually not only susceptible to one type of maintenance, having a large variety of systems and parts with different inspection and replacement periods. This creates a problem of possibly having multiple disruptions on its operation, which had a solution first presented by [36] with the introduction of the concept of "opportunistic replacement", where a part of a system might be replaced even though it has not failed or reached its end-of-life, using the period of maintenance required by another piece of the system, if this results in reduced over-all costs as it reduces non-operational times.

However, in order to represent the system in a realistic manner through the use of mathematical models, a large number of simplifications and assumptions is necessary, rendering the solution unrealistic. [14], for example, analyzed the case of Air Canada's maintenance planning in the 70's and came to the conclusion that finding an exact solution, using a Mixed-Integer Program for examples, was not a feasible method, as it could not coupe with the complexity of the problem. Furthermore, [14] argues that even if a viable optimal solution was possible to be achieved, the high unpredictability of an airline's operation could rapidly turn the solution unfeasible and, thus, finding a good solution is sufficient. To do so, the found solution was to develop a simulation platform called AMOS, where the main parameters can be inputted in order to allow for the user to visualize and optimize the maintenance scheduling program.

The restrictions and assumptions needed to model the maintenance scheduling process as linear expressions, as seen, led to different approach to model such problems, mainly leading to the use of simulations, them being supporting tools such as AMOS or computational prediction tools, such as Monte Carlo simulations [46]. This left the maintenance scheduling process with three main modeling methods, through the use of simulations, linear functions and with a Markov Decision Process (MDP). Methods to search for optimality of these models vary greatly and have many applications throughout all industries.

3.3. Engine Shop Visit Scheduling process

The research onto engine shop visit scheduling optimization, being a very specific topic, is limited. Just as in the case for general maintenance scheduling, ESVs are currently frequently manually scheduled. Among the available research, once again the problem is modeled mostly modeled through mathematical equations or through simulation methods.

3.3.1. Simulation based

Different simulation tools have been developed to simulate the engine shop visit operation and scheduling. Simulations are most frequently modeled as a decision making support tool, leaving the scheduling optimization to the operator, as in the case for [54] and [48], or providing some level of optimization, as presented by [14].

Decision making support tools

These tools usually seek to integrate engine data, environmental and operational conditions, maintenance strategies and MRO capacity as inputs to provide maintenance performance metrics, as for example mean time between removals, shop visit cost, number of necessary spare engines and others. [54] uses a discrete event simulation model and is mathematically defined in Matlab to estimate ideal ESV dates, engine degradation with time, engine overhaul turn-around time and other, using flight cycles as time interval in an engine life-cycle span. It considers LLPs, performance and random failures as ESV triggers, using a total of 9 items that can be potential causes. [48] also uses multiple operational, engine performance and contractual data as inputs to mathematically predict LLPs end of life moments and performance deterioration to determine ESV times and maintenance costs. It then creates a base of different scenarios to be evaluated by the operator using a iterative algorithm model, setting different thrust settings, SV schedule or workscope definitions and searching for lower total maintenance costs. The presented tools allow for the comparison between the effects of different maintenance policies and schedules on maintenance costs and spare parts planning, helping management in the decision making process. Such supporting tools, however, dont aim at finding the optimal shop visit scheduling, rather helping experienced professionals to execute a manual task in a manner of hours, instead of days.

Multi-agent negotiation

One strategy to tackle the optimization problem inside a simulation environment is through the use of a multi-agent negotiation method, as used in [53]. In accordance with [14], the authors believe that reaching an optimal solution is in practice not a requirement, being sufficient to find solution that improves productivity. They model the problem with 3 agent types: the overhaul bases, the fleet managers and the fleet planner. Maximum shop visit times are predicted by the fleet planner to determine the date of zero life to LLP parts, the Date of Zero Margin (DOZM) for parts and the date of zero confidence for the predicted time a parts failure risk achieves a set threshold. Dates can change depending on service data and unforeseen events. The overhaul bases may request different schedules due to its operational necessities. The scheduling algorithm was implemented in Java and attempts to maximize cost reduction by swapping engine shop appointments and set them as close as possible to the latest removal date, negotiating with the requirements set by all agents. Importantly, the model considers the uncertainty of the maintenance turnaround time, correlating it to the number of spare engines, fleet size and maintenance center capacity. The model calculates maintenance costs as a function of the time distance to the estimated maximum maintenance date, neglecting the possibility of opportunistic replacement and also not considering the possibility of using the AoG moment caused by the aircrafts other engine as an ESV opportunity.

3.3.2. Optimization based

The optimization tools are developed with the purpose of providing an optimal, or close to optimal, solution to the maintenance scheduling of engine shop visits. One method frequently used to simulate the engine shop visit scheduling process is modeling the problem through a Markov Decision Process (MDP). The optimization can be also modeled as a mathematical problem, being a solution possible by solving a set of equations.

Mathematical model

As seen, opportunistic replacement is an important concept for the optimal maintenance of a multi-state system. [7] aims at mathematically model the opportunistic replacement problem and uses a jet engine maintenance scheduling as a possible application for their model. They model the most basic case, where a set of components have a defined maximum time to be replaced, considering a limited set of time steps, a cost of replacement and a cost of maintenance, which is independent of the number of items being replaced. The mathematical model consists of an objective function, defined by the sum of replacement and maintenance costs, and 6 constraints, defining the variables and guaranteeing the replacement of the items before their end-of-life and the payment of the fixed maintenance costs. These constraints alone, however, are not sufficient to model the problem as a convex hull, being necessary the derivation of new ones using ChvátalGomory rounding. If costs can be considered non-increasing with time, the problem is proved to have a single optimal solution, instead of many, and can be solved by a dual greedy procedure. The model was implemented in Python and solved by the mixed integer programming solver Gurobi. To test the model, a case study using the scheduling optimization of engine shop visits is performed for two parts of an engine. Firstly, considering the low pressure turbine and its 10 components, the obtained optimal solution is compared to a non-opportunistic approach, where every time a component reaches its end-of-life a separate maintenance procedure is scheduled. The results show a reduction of almost 70% in the amount of maintenance occasions and 34% reduction in total maintenance costs. Similarly, the same analysis is performed for the high pressure turbine and its 9 components, also achieving improvements in costs and number of maintenance occasions, although with reduced significance.

Another strategy to consider LLP and EGT degradation together is to consider a LLP as a normal distribution with very small variance. In fact, when considering that LLP are set in cycles (or flying hours) and an airplane might have different operational periods, flying a variable amount of cycles in a day, converting the number of cycles up to a set calendar date present uncertainties, resulting in a value with some variance. [19] uses this strategy in conjunction with a just-in-time (JIT) concept for the engine repair. It does that by considering that only one spare engine is necessary if the engine shop visits of an aircraft are scheduled in such a fashion that once one engine returns from the shop visit, the AoG opportunity is used to remove the other engine, which in turn is also sent for a shop visit, and the spare engine swap wings. To model the engine degradation, the EGT margin is described as a State Space Model. A Bayesin state estimation and prediction method is used to obtain the cumulative distribution function and probability density function to determine the probability of the engine reaching a defined EGT threshold for each time step of the analysis. Considering also uncertainties on the time required to perform the maintenance in the engine at the shop visit, the paper mathematically defines the optimal point for the first engine shop visit based on the maximum acceptable probability risk of failure on the first engine and the acceptable maximum probability of shortage of the second engine due to its RUL, that is because in the case that the second engine needs a SV before the first engine returns from its SV, a urgent spare engine is necessary, resulting in much greater costs. The concept can be better visualized by Fig. 3.1. The paper concludes that the SSM can better predict the engines RUL when compared to a simple linear regression and that its optimization method has a feasible implementation by operators. This paper, however, considers a single engine, and not a fleet of multiple airplanes and the possibility of using engines of other aircraft returning from SV as spare engines. More importantly, the method assumes a generalist RUL probability distribution, independent from the different possible causes for ESVs. Furthermore, the optimization process considers EGTM and LLP separately, finding the ideal ESV for each factor, which in turn disregards the possibility of opportunistic replacement. At last, the authors assume the absence of random failures.

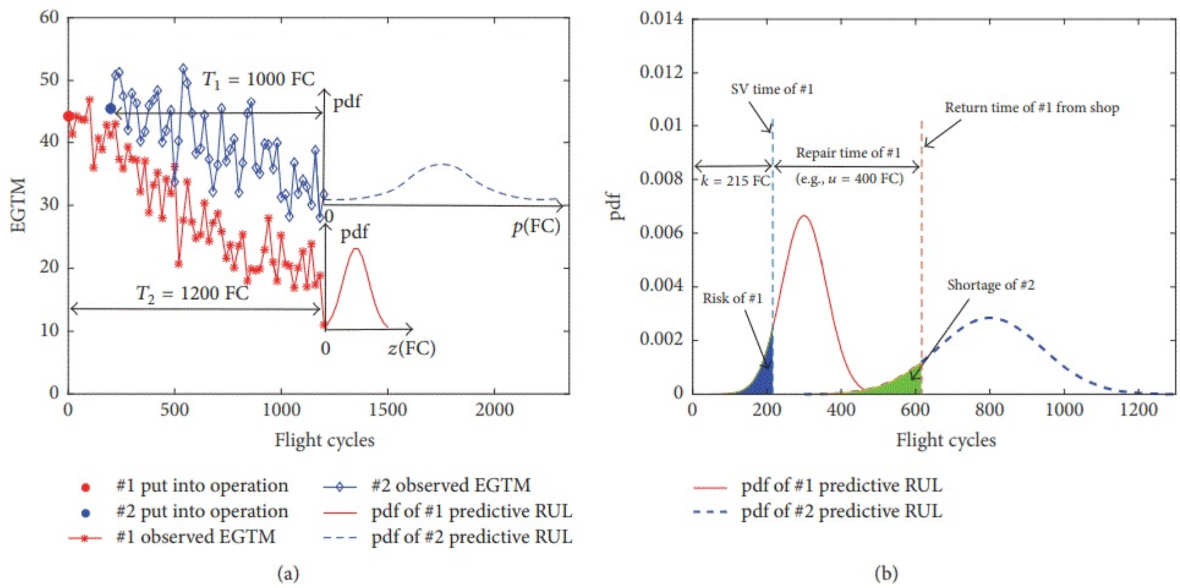


Figure 3.1: Definition of optimal SV point from failure pdf. [19]

Markov Decision Process

As engines combine condition-based maintenance, hard-time maintenance and corrective maintenance items, a hybrid approach is necessary to try to optimize the engine shop visit scheduling considering all factors.

MDPs allow for this hybrid representation, presenting the system as a state vector that can represent the different components requiring maintenance in an engine.

[43], aspires to optimize maintenance intervals and maintenance workscopes in conjunction through the use of a reinforcement learning approach. To do so, the engine is represented by a multi-dimensional state space. In the model, an agent chooses a maintenance task in response to the engines state. LLP parts usage and EGT degradation are modeled with a Markov Decision Process and failures are considered in the state transition, where the failure probability is correlated to the engines performance state. The reinforcement learning is performed with a GaussSeidel value iteration algorithm, used to explore different possibilities of maintenance policies, iterating over different schedules and workscopes and searching for the sequence of maintenance tasks which minimizes the total cost. As it considers different workscopes, it considers the possible benefits of opportunistic replacement. This analysis, however, does not consider maintenance periods and extra costs involved with the use of spare engines and operational costs related to the time needed to reinstall the overhauled engine on its original aircraft. In it, maintenance actions are considered instant, changing the engines state from one time step to the next subsequent step.

Hybrid maintenance policies can also be approached by other methods. [20], for example, uses genetic algorithms for the optimization process. In it, 4 different maintenance items in a system are analyzed considering their interdependencies. Moreover, [20]s approach is to not use prognostics to predict the approach of thresholds, rather to predict total system failure probabilities and use them in conjunction with inventory levels to optimize the maintenance scheduling. System risk is defined as a sum of failure costs and maintenance losses, which in turn are modeled with mathematical equations and have their failure probabilities obtained from reliability data with a Weibull distribution. This can be correlated to the study case, where 3 different items (considering EGT performance, failures and the grouping of LLP items) are considered together to optimize the maintenance periods. Maintenance schedules are represented by a set of binary strings, which are then evaluated for their total system risk. Using GA, a schedule solution which minimizes the total risk is found. The author goes on to compare the obtained results with the results that would be obtained by implementing the traditional CM, PM and CBM (with thresholds) separately, achieving better results with the proposed use of failure probabilities in combination with genetic algorithm. The research, however, doesnt consider the optimization of a combination of multiple systems, as it would be the case in a fleet with many engines, and how maintenance opportunities in one system may present opportunities for another system, as for example when an aircraft is put out of operation for the re-installation of its original engine, the other engine could be removed with reduced operational impact (one less AoG period). Furthermore, as in the previous study, it considers maintenance to be instantaneous from one time step to the next, when performed.

[33] provides a different approach to use genetic algorithms to optimize an individual engine shop visit scheduling. Instead of a binary string, the proposed method uses real-valued strings in a pair of chromosomes, with the first half containing to the number of cycles up to the shop visit and the second half containing the work scope of the corresponding gene shop visit. The paper compares two different logic methods for the generation of the initial population of chromosomes for the implementation of the genetic algorithm optimization process. The engine shop visit problem is analyzed in the context of the engines entire life cycle and only considering LLPs and the aircraft maintenance requirements. Maintenance work scopes are divided between 3 levels of maintenance, from minimum to heavy, and 4 maintenance actions, from no LLP replacement to both LLP types replaced. The objective function is set in order to achieve a minimal total maintenance cost per flight cycle. The fleet is not considered, nor the use of spare engines, the concept of just-in-time maintenance or other ESV triggers, as EGTM and unpredictable failures.

4 Maintenance Scheduling Problems

Although the engine shop visit scheduling has its particularities that have to be taken into consideration for a useful and successful implementation, it is just a specific case of the general scheduling problem. This type of scheduling problem can also be seen in other aviation and operations related topics, as for example in the maintenance of the aircraft as a whole. Although engines are a part of commercial airplanes, the maintenance policies of engines and the aircraft as a system are separate and have distinct frequencies and thresholds. Aircraft maintenance tasks are, as a standard, grouped in 4 categories: A, B, C and D checks. C and D checks are considered heavy maintenance and happen less frequently, requiring a long term planning [58], as it is the case of ESV scheduling.

A comprehensive literature review on the most recent (up to the time of publication of the article, in 2013) operations research applied to aircraft maintenance is provided by [57], dividing the articles into 6 different subcategories: Type of maintenance; Integrated airline scheduling; Aircraft maintenance optimization; Facility location; Workforce and training; and Uncertainty and application of research.

While from the literature review it is clear that, for aircraft and engine maintenance, the use of simulations and MDP are the main approaches to model the scheduling problem, another type of formulation can also be seen when looking into other maintenance scheduling activities. According to [39], scheduling problems are commonly divided into three different shop problems: open shop, flow shop and job shop.

This chapter seeks to answer the sub-question ”What are the most used methods to model other maintenance processes scheduling?" and the sub-sub-question ”What methods are used for the optimization of other maintenance scheduling problems?".

4.1. Dynamic Programming

[25] proposes a method to optimize the long-term planning of C checks while also considering A checks, aiming at reducing the number maintenance occasions and increase aircraft availability. The problem is mathematically modeled in a framework prepared to be solved with dynamic programming, with an objective function which seeks to minimize the number of remaining hours until necessary maintenance at the time of the scheduled procedure. The authors present the unavailability of reliable cost data as reasoning for not using cost reduction as optimization objective, as well as the direct correlation between number of checks and total life time costs and the high costs of AoG in comparison to the costs of performing the necessary maintenance. To model the problem 16 constraints are set, considering the hangar capacity per day for each type of maintenance and allowing for tolerance interval, although with a high penalty (with tolerance interval, airlines may perform maintenance after the maximum stipulated limit, but have to compensate for it in the next check scheduling process). The process is defined through a decision space, a state space and a state transition, following the structure of a Markov Decision Process, as it was also frequently seen in ESV scheduling. An unbounded approach to solve the problem would create an exponentially increasing number of outputs, leading to a dimensionality problem. For example, by having only 10 aircraft and two maintenance slots per day, in three days a total of 1.7 millions outcome possibilities have to be analyzed, thus being computationally inefficient. Trying to reduce the amount of paths to be evaluated, the authors propose three additional steps: giving priority to the aircraft that is closer to its due maintenance date and require a C check, as it is more restrictive; doing a thrifty analysis on every possible outcome and eliminate the ones that

might result in an aircraft on ground waiting for a maintenance opportunity, which is undesirable; bounding the number of outcomes by discretizing the problem in respect to the fleets utilization and then aggregating them into mean utilization and electing one of the outcomes to represent the entire group. The implemented method is successfully able to optimize in minutes the maintenance scheduling for years. Comparing it to the manual scheduling from a European airline, the method is able to reduce the number of maintenance checks and, consequently, reduce costs and increase aircraft availability.

[50] introduces the use of dynamic programming to the case of scheduling maintenance for systems subject to stochastic failure, as aircraft engines for example, by optimizing it to reduce total maintenance costs, considering the system‘s downtime; [42] uses DP to solve the Airline Maintenance Scheduling Problem, defining which airplanes should be maintain and which maintenance tasks should be performed on a daily basis; [17] uses a MDP to model a multi-state deteriorating wind turbine and optimizes its maintenance strategy with a backward induced DP, considering Condition-Based Maintenance and including seasonal effects on the turbine‘s state; [62] models an opportunistic preventive maintenance for a multi-unit system and finds the optimal solution through DP, considering imperfect maintenance (maintenance doesn‘t return the parts to the same condition as new) and using total costs as objective function.

4.2. Mixed-Integer Linear Programming

As an engine comes for a shop visit in an MRO provider, it is disassembled into multiple sets that have to go to different maintenance procedures. [21] uses a variation of the flexible job-shop problem, allowing for an operation to be performed in any machine in a set of them, adapting the MILP formulation first developed by [45]. The authors use the tardiness of a job as objective function, aiming to provide a more customer-oriented approach to the problem, as the tardiness refers to the time difference between the scheduled maximum due date and the actual completion time. As a constraint, it is considered that some operations can only occur after the realization of another. The final model has 10 constraints and was implemented in AMPL and optimized with Gurobi. By using data from a major European airline, the authors optimize a set of engines going to maintenance based on a initial set provided by the airline, they are, however, unable to optimize the full scenario for an entire week due to dimensionality problems, limiting their analysis to 8 engines with a total of 179 parts.

4.3. Genetic Algorithms

[25] is also the base for the methodology proposed by [58], combining a minmax scenario with genetic algorithms to search for the optimal solution. With the minmax case, an optimal solution for the worst case scenario is found, which is in turn a feasible solution for all other scenarios. Scenarios are created using Monte Carlo simulations and consider two factors as having uncertainty: the duration of the maintenance procedure, being the critical cases the ones which have a much shorter or much longer maintenance period in comparison to the average value; the daily aircraft utilization, being the critical case when the aircraft has a much higher utilization than the average, thus requiring maintenance in a shorter time span. The daily utilization of the aircraft is represented with a vector size 12, representing the average aircraft utilization for each month in a year, allowing to capture seasonal variations frequently seen throughout the year. Differently from [25], [58] formulates the maintenance problem as an integer linear programming problem. For the optimization, the schedule is modeled with a chromosomic representation, where each line represents a different aircraft and each column represents the number of time steps from the simulation start point to the according maintenance procedure. The initial population is created using an ϵ-greedy algorithm, which not necessarily will generate feasible schedules. The set is evaluated with a fitting function equal to the objective function, and as in the case of [25], evaluates the number of remaining hours available at the time of maintenance. As the problem is modeled as an ILP, it could theoretically also be solved with a commercial LP solver, however the scale of the problem makes it computationally impractical, as each aircraft has a decision variable for each time step, presenting dimensionality problems as also seen in [25]. To validate the GA model, a small scenario was created and compared to the LP solution. For cases with a low amount of aircraft and short planning period, both methods reach the same solution at comparable computation times, however considering 40 aircraft and a short 3 year time span, the LP solver takes more than 60 times the computational time of the GA model, while the GA is able to keep a small optimality gap. When comparing with the DP method presented in [25] for a deterministic scenario, the GA optimization is able to increase the average number of flight hours between maintenance and reduce the overall number of C-checks performed. When using the minmax scenario, it is able to reach a significantly better result then the manual airline scheduling

process in both flight hours and number of C-Checks, at the same time that it generates its solution has a high robustness, not requiring any changes in 40% of cases created with a Monte Carlo simulation, compared to 0,27% in the deterministic optimization.

When considering preventive maintenance, two types of maintenance can be performed, simple preventive maintenance and preventive replacement. [56] considers both options, and their impacts on a general decaying system, to calculate with GA the best scheduling and policy in regards of total life costs for the system. Considering a manufacturing system, [61] considers both the gains of production on a well-maintained machine and the maintenance expenses, in a system with increasing failure probability with time. Each cell in the chromosome represents one machine and has an integer value, corresponding to the time of maintenance. Probability of failure is evaluated using Monte-Carlo simulations and then used to estimate total life-cycle costs during the GA optimization. Results show a reduction in costs when compared to simulated FBM, TBM and CBM methods. [59] applies a maintenance scheduling optimization to increase the reliability of a power generation system with many generators, by having as objective goal the minimization of the loss of load expectation, that is, the amount of days in a year that the system wont generate the expected output. The method was tested using real life power system data and observed improvements on loss of load using GA. Importantly, it also saw a higher impact of the initial population on the final result for a smaller number of generators, with a reduced impact on larger systems, where a standard initiation procedure with random chromosomes will produce similar end results.

[39] has a similar objective as the one seen in [21], searching for the optimal way to distribute and sequence the 5 parts of an aircraft engines that arrives for maintenance among their correspondent work areas. They increase the complexity of the problem by also analyzing the mean time to repair each of the parts and, in the case of a large difference between them, consider the possibility of swapping parts among engines, thus modeling the problem as a job shop problem within a flow shop problem. The optimization is performed with genetic algorithm, using a unique chromosome representation, as described in Section 5.5. The goal of the optimization is to minimize the time that an engines stays in the shop and the number of times the engine needs an ESV. To do so, two fitness functions are used: First the makespan for each engine is calculated by adding the repair time, the swap time and the test time; secondly, it calculates how many components have a mean time to repair out of the set tolerance and, thus, will be swapped. The optimization in done with GENMOP, a Pareto-based algorithm that utilizes real values for crossover and mutation operators, according to the authors, achieving what they consider as good results, however limited to a small number of engines.

4.4. Reinforcement learning

The method proposed by [25] was further explored by [8] by applying a reinforcement learning (RL) approach to optimize the problem. This was done by using Deep Q-learning with experience replay and a Double Q-learning variant. In comparison to the DP solving approach, the RL method was able to reach a schedule with a lower average stub life value for C-Checks and only needing seconds to find a solution, albeit a long time is necessary for the model training, in the presented case it was 20 hours. The authors also show the flexibility of the proposed solution by implementing disturbances into the initial data, as once the model is trained, it takes a minimal amount of time to find a solution in case there are changes in the base scenario.

The JSSP has been extensively studied and multiple optimization approaches have been evaluated. As seen, among the most recent approaches genetic algorithms is once again used, however other techines are frequently used well, such as branch-and-bound algorithms, beam search, simulated annealing, tabu search and ant colony optimization [[39],[45]]. [29] evaluates the use of one of the most recent solution approaches, reinforcement learning. To model the JSSP problem, a multi-agent Markov Decision Process (MMDP) is used, where each resource has an agent attributed to it. An MMDP allows for local decisions and reacting scheduling, so that the model can more easily adapt to unpredictable events. Using a Multi-Agent Production Scheduling framework (MAPS), a decision vector is updated by the corresponding agent, where each decision occurs after a previous operation is ended. The authors present a difference between joint-action learners and independent learners, where in the first type agents are aware of their state and contribution to the system as well as the state and contribution from all other agents, where in the later each agent has only access to their own set of information, being the later the system used by the MAPS framework. To train the model, a Q learning method is implemented, deriving a control policy from a limited (and optimistic) training set. The method is compared to traditional methods for classical operations research benchmarks.

Many other articles tackle the scheduling optimization problem using reinforcement learning. [24] unites preventive maintenance into production scheduling using reinforcement learning using a MDP model. They propose a novel model-free algorithm and compare to traditional methods achieving encouraging results. In [6], reinforcement learning is used in a petroleum industry application, by considering the dynamic scheduling of maintenance tasks problem, modeled with a MDP and solved with a State-Action-Reward-State-Action algorithm, instead of the Q-learning, thus using an on-policy learning approach. [22] applies reinforcement learning to schedule real-time workloads received by a cloud services provider, aiming at reducing virtual machine use costs while keeping the quality of service high for the client, suing a Deep Q-Learning.

4.5. Discussion on maintenance scheduling models

This section provides a discussion upon the presented methods to model maintenance scheduling processes and how they can be used in the ESV scheduling case. The objective function is also discussed, as part of the answer to the sub-question "What factor influence the cost of an ESV?".

4.5.1. Analysis

As seen, it is clear that the scheduling optimization problem has been widely researched since the 60s as part of the operations research field. The engine shop visit scheduling problem is a subset of this wider problem, having particularities which make it unique and complex.

As these are complex problems, many different ways to model them were created, as seen in Chapter 3. Modeling seems to resume to simulations and mathematical models. When considering simulations, most approaches aim at providing a decision support tool, with none or minimal computational optimization. Looking into mathematical models, it is seen that there are many ways to model the problem, from using probabilities and time periods to find a system of equations that can be solved, as in [11] and [19], to more complex MDP models, as seen in [[43],[20],[33],[25],[29]].

To this day, one of the favorite methods for operators to schedule their maintenance operations is doing it manually with the assistance of simulation platforms. While simulations are a helpful and intuitive tool to support decision making from aircraft operators, they dont guarantee that the solution will be robust nor close to optimal, only guaranteeing feasibility at the moment of scheduling. The scheduling process is also longer and may take days for the operators to find a feasible scenario and try to find ways to improve it. [[54],[48],[14]] show more modern approaches to produce decision making supporting simulation tools, predicting end of life of LLP parts and degradation of performance up to a defined threshold, and even trying to perform optimization with multi-agent negotiation methods. As advantages, this type of simulations present a very intuitive operation an allow for a easy performance of a manual sensitive analysis, changing scheduling and looking on how this affect the costs. It allows for quick changes and usually have a visual appealing interface for the operator. On the other hand, the short comings of decision making support tools are obvious, being the main points the long time needed to do the scheduling, especially for larger fleets; and the not optimal schedule, limiting it's function to find feasible schedules.

When aiming at optimizing the scheduling of ESVs, it is seen [[43],[20],[33],[7]] that most methods focus on a single engine scheduling. Opportunistic replacement plays key role in scheduling optimization, as it allows to consider the impact of having the system down for a period of time and how this may make beneficial the performance of maintenance tasks in parts that might still have considerable amounts of stub life on them, but would require the system to be non-operational once more. This is precisely the case for engines, as there is a multitude of sub-systems of an engine with different due dates and a high cost of grounding an aircraft for maintenance. In this case, two focus points can be identified as being considerable for the opportunistic replacement implementation, them being the LLP limits and the engine degradation pattern represented by the EGTM. The moment where an ESV will be required to perform LLP replacement can be estimated by evaluating the pattern of cycles per day for a set aircraft and performing a simple linear regression. To estimate the moment where an ESV is required due to EGTM can be, as seen, more complicated, as the degradation does not follow a linear pattern. In some cases, as for example in [43], performance degradation is considered in a more complex way, defining it as a state in a MDP and defining degradation probabilities. Cases as the one seen in [19] go one step further, modeling the performance degradation as a state space model, considering observation noise and the different rates of performance degradation rate throughout the engine's operational life. Their results, however, show that the degradation pattern correlation coefficients indicate a close resemble to linear degradation, and although SSM results show better precision than linear regression, the difference is not that significant. Thus, it seems that for a life cycle management estimation, a linear regression considering the period of stable degradation of the engine, after the initial wear period, is enough to provide a long-term estimate.

It is also clear that this type of problem is frequently modeled as a Markov Decision Process, using different optimization methods to search for the optimal schedule. Most frequently, however, MDPs focus on the modeling of the particular engine and are modeled to update the state of it to fully restored as soon as the action of performing maintenance is selected. In reality, maintenance on the engine takes long times, during which the system (in this case the aircraft) will be operational using a spare engine. This interrelationship between aircraft and engine, and the costs correspondent to the state of each of them, brings an additional complexity to the problem. For a practical application, both have to be modeled and considered together in the optimization. Among the optimization methods the most seen are: mathematical solution (including linear programming solving methods), negotiation methods, genetic algorithms and reinforcement learning.

The scheduling of aircraft maintenance presents similar long-term planning challenges as for the schedule of ESVs, thus also providing valuable research for the scheduling problem. In fact, when looking at the research available on the topic, similarities are visible on the modeling structure of the problem, again mostly seeing the use of Markov Decision Processes, but also seeing the use of the Job-Shop Scheduling Problem (JSSP) for some cases. For the solving process, again LP solving, Dynamic Programming, Reinforcement Learning and Genetic Algorithms are most frequently used. The shortcomings are also similar, encountering dimensionality problems when considering a large time span and a larger fleet and trying to find the optimal solution. As these are long term planning scenarios, for over 10 year periods, having a time step of a single day will result in a very large number of decision variables, thus a solution might be reduce the time precision to longer steps, as a week for example.

It is clear that research has been mostly focused on modeling long-term planning schedules as MDP, leaving the JSSP for more short term and more standard problems. [29] brings a rather unique approach to the problem, uniting the JSSP with a multi-agent MDP and reinforcement learning to solve it. This is an approach that could be adapted to the ESV scheduling problem, where each engine or each aircraft is modeled as an agent and solved to optimize the total costs for the entire fleet. The use of aircraft as agents allow for the cost considerations regarding Just-in-time engine maintenance, as introduced by [19], as well as optimizing for the entire fleet. The consideration of the fleet is important as engines might be swapped among all aircraft, allowing for a higher flexibility in the model.

At last, although many researchers use remaining hours as a minimization objective function, aiming at using all the parts for the longest possible time, this might not be the ideal approach for an airline seeking to reduce costs. Although it is evident that using parts as much as possible increases the return on investment, it ignores the possible financial gains of using opportunistic replacement to perform maintenance on other parts and the costs (and loss of revenue) for each time an AoG is necessary. Thus an evaluation of total costs seems more appropriate for practical applications. Furthermore, sensitivity analysis is mostly not evaluated in the presented research, but might be of high value for airlines, as they try to evaluate different scenarios, as with the possibility of leasing extensions, for example.

5

Optimization Methods

This chapter goes deeper into the main modeling frameworks and the optimization techniques used to solve the problem, presenting the main methods most frequently seen in research. It seeks to expand the answers related to the sub-questions "What methods are used for the optimization of ESV scheduling?" and "What methods are used for the optimization of other maintenance scheduling problems?".

5.1. Linear Programming

According to [31]: linear programming involves the planning of activities to obtain an optimal result, i.e., a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives. Any problem that can be mathematically described with linear equations as an objective function and constraints is a LP problem, being thus a formulation method. To solve this problem, an array of procedures are available. The most common method is called the simplex method. The simplex method is an algebraic procedure based on geometrics. As the constraints create solution boundaries, if these constraints are plotted in a graphical manner, it is possible to see that it creates a solution area, where the corner-points are the possible candidates for the optimal solution. Other solution techniques are also possible, as dual simplex, parametric linear programming, interior-point algorithm and others, as heuristic solutions for example, which dont guarantee an optimal solution. These methods are then used by commercial solvers, like Gurobi and CPLEX, to find the optimal solution, as it was the case in [7] and [21]. [21] models the problem as a Mixed Integer Linear Programming (MILP), a variation of the LP problem, where some of the decision variables must be integers, differently from the standard LP problem, where they can be decimals and the Integer Programming, where all of them have to be integers.

On a JSSP, for example, the problem can be modeled with binary variables indicating if a job i is assigned to a machine j at the t, in the form of $x_{ijt} = 1$ in case it is and $x_{ijt} = 0$ in case it is not. Constraints can be set so that a job is always assigned to a machine and that a machine is not assigned to more then one job at the same time, among others, depending on the constraints of the problem. Variations of this base case can be seen as in the case of [21], where one decision variable is a binary indicating if a job is performed on a machine and one binary decision variable to order the operations, indicating if a job is performed before another or not. Other articles that also are modeled as MILP also use similar formulations: [58] has 5 decisions variables, indicating if a check occurs in an aircraft at a set time, the number of flight hours and days since the last check, the existence of extra maintenance slots at a certain time, if aircraft are under maintenance at a time and if at a set time at least on aircraft is in maintenance. In [7], the base case of opportunistic replacement, only two decisions variables are necessary, both being binaries and indicating if a maintenance occurs at a certain time and if a set component is replaced at that time.

5.2. Markov Decision Process

As described by [34], one of the most important tools to model and solve sequential decision-making problems is the Markov Decision Process (MDP), and have been used in many practical applications. MDPs are divided into three main types: Discrete time MDPs, continuous time MDPs, and semi-Markov decisions processes. The most basic case is the discrete time MDP, where an agent interacts with an environment in discrete time steps. This process can be defined as a tuple [51]:

$$M = (S, A, T, R) \tag{5.1}$$

where S is a finite set of states that define the environment. It is assumed that the complete state is always known at each decision moment and provide all information necessary to reach a decision. A is a finite set of actions, containing all actions that can be performed for a set state. T is a transition function, that provides the probability of reaching a following state s', given the current state and the action chosen. R is the reward function, providing an expected value consequence of choosing an action based on all possible outcomes from it.

The result of the MDP is a policy π, that defines which action should be taken knowing the state that the agent is located. An optimal policy will lead the agent to a path of optimal decisions which will maximize the reward of the system. In a Markov policy, actions are chosen without consideration of the system's past. If at a defined state a fixed action is always chosen, it is considered a stationary policy. According to [15], for a finite-state and finite-action problem, an optimal stationary policy will exist.

The expected cumulative reward for a given policy at a certain state can be obtained by a value function. An optimization process will try to determine the policy with the highest value function from the starting point of the system. The number of possible combinations of states and actions might become very large as a larger number of states, actions and time steps are evaluated, depending on the problem. Thus, many different methods to try to optimize the MDP policy without experimenting all possible combinations are considered.

5.3. Dynamic Programming

Dynamic Programming is a multi-stage decision process first conceptualized by Richard Bellman [13]. A multi-stage decision process is where a system is defined by a set of states that change in time according to the result of a sequence of decisions. When the decision of an action incurs in sub-problems, and a same sub-problems might appear as a result of different possible decisions, then this overlap of sub-problems allows to reduce the size of the general problem by only solving this set of sub-problems and then combining their results in a bottom-top approach to obtain the original problem‘s solution [23]. It does that in by making a time-memory trade-off, saving previous results and looking at each step if the current sub-problem was already solved in a previous step. This is based on the principle of optimality [13], which states that if the optimal solution for a sub-problem depends only on its current state and not on any past history, then the original problem‘s optimal solution can be obtained from the merge of the sub-problems‘ solutions.

This ability makes dynamic programming a powerful and widely used optimization tool, significantly increasing efficiency and reducing processing times when compared to an optimization solution by looking at all possible combinations.

Moreover, Richard Bellman's work on dynamic programming laid the foundation for solving complex sequential decision-making problems involving uncertainty, known as Markov Decision Processes (MDPs), as previously seen. The Bellman Equation, introduced by Bellman, plays an essential role in solving MDPs. It expresses the value of a state in terms of the expected immediate reward and the value of the next state, facilitating the determination of optimal policies. Algorithms such as Value Iteration and Policy Iteration build upon the Bellman Equation to efficiently find the best strategies for navigating MDPs, which will also be essential concepts for Reinforcement Learning.

On [25], as the final state is not known, a forward induction DP has to be implemented, where for each sub-state, the shortest path from the initial state to it is computed. Due to the number of possible sub-states, this creates an unpractical method to be computed, requiring the previously mentioned size reduction actions. The optimal schedule can be then obtained by recursively computing Bellman‘s equation:

$$V_t(S_t) = min\left\{C_t(S_t, X_t) + \gamma \sum_{S_{t+1}} p(S_{t+1}|S_t, X_t) V_{t+1}(S_{t+1})\right\} \tag{5.2}$$

5.4. Reinforcement Learning

Dynamic programming is also the base for another method of problem solving. One of the main ways for humans to learn anything is through interaction with the environment, analyzing the correlations between action and consequence. According to [[37],[55]], this is the base idea for the development of reinforcement learning, where an agent learns by trial-and-error in a defined environment. In it, the path to reach the goal is not clear. On every step, the agent chooses an action which affects the current state of the environment, this effect is communicated and reinforced to the agent. The main goal of the agent is to choose actions that will lead to an improvement in the final value for the objective function, without previously knowing which action to take and what the result will be. Typically the environment will be non-deterministic, meaning that a defined action in a particular state will generate different results depending on the moment it is executed.

The dilemma for the agent is to prefer actions that it knows from experience that are good, performing exploitation, and also search for other options to try to find new better paths, performing exploration. It is impossible to perform only exploration or exploitation and be successful on its task, thus, a good trade-off between them has to be achieved. Even with a good trade-off, reinforcement learning does not guarantee a global optimal solution, rather it will find sub-optimal solutions.

The algorithm thus has to repeatedly update its actions for defined states until convergence to a best decision policy is reached. This is done with by a value iteration algorithm or a policy iteration algorithm. In the case of [43], a Gauss-Seidel value iteration algorithm is used, taking into consideration previous decisions in nearby states and, thus, increasing the efficiency of the algorithm, at the risk of not reaching convergence. For Gauss-Seidel to be implemented, however, it is necessary to know the environment‘s model, as transition probabilities and rewards. This necessity is avoided by using a model-free algorithm, as Q-learning for example, used in [8], as the cost of requiring a larger exploration and, thus, processing times.

[47] also provides a reinforcement learning approach to solve the flexible job shop scheduling problem in general terms, using Q-Learning. They divide the optimization process in a two-phase learning process, first determining the most fitting machines to each operation and later only determining the sequence of the operations, already knowing the ideal machine, with the objective of minimizing makespan. On [28] a comparison between the use of deep reinforcement learning (with a proximal policy optimization) and genetic algorithms to find an optimal solution for a FJSP is performed, also using as objective the reduction of makespan. They conclude that RL is able to find close to optimal results in short processing times and outperforms GA in most cases. [16] goes one step further, considering the possibility of machine breakdown on a FJSP using another type of machine learning, with a teaching-learning optimization method. In doing so, instead of only looking at makespan, as was the case in the other presented cases, robustness and stability of the solution are also evaluated and taken into consideration in the second part of the optimization.

5.5. Genetic Algorithms

Another optimization method based on natural processes is Genetic Algorithms (GA). In GA, a combination of crossover, mutation and natural selection is computationally implemented to find the fittest result in an evolutionary process. The process begins with an initial set of solutions (the initial population), usually randomly created and covering a large part of the scenario space. In every step the state of the system is represented by a vector or a string, comparable to a chromosome, where every gene in the chromosome is a value (usually binary or integer). At each iteration, a set of chromosomes will be selected, based on their evaluation to an objective function, and combined to create offsprings. This combination is done with crossover, where each parent will be split in a random position and combined. Offsprings go through a mutation process, where the new chromosome might have some of its genes randomly changed to a different value, based on defined probabilistics. This process is visualized in Fig. 5.1. The offspring will set the new population for further iterations, however before a new iteration begins it is necessary to analyze their fitness, that is, the quality of the solution to the objective. Just as in the case with reinforcement learning, GA don‘t guarantee a global optimal [[30],[40]].

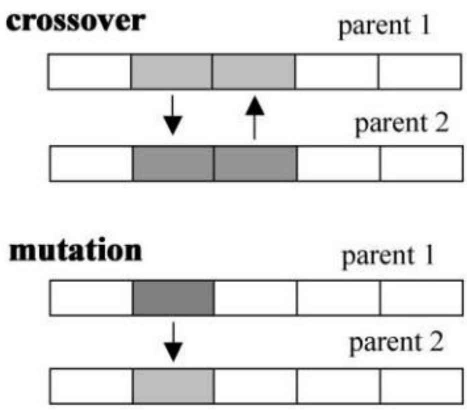


Figure 5.1: Example of crossover and mutation processes. [56]

In [20], chromosomes are defined by a string with 200 binary values, correspondent to each of the components for each of the 50 time steps. An example with integer genes is given by [33], where chromosomes are set up in accordance to the number of engines and the number of shop visits, where the first half sets to the number of cycles up to the maintenance task and the correspondent gene in the second half sets the workscope of the maintenance, among 7 different work levels. They also compare two different methods for the initial populations creation. Another research using a variable sized chromosome representation is [39], where a direct encoding approach is used with the first part of the chromosome presents all engines in order of priority and the second part specifies the precedence of the component exchanges, the quantity of exchanges to be made, the specific components involved in the exchanges, and the engines from which the components will originate. At last, on [58], each chromosome represents one aircraft and each gene has an integer representing the number of cycles up to the next maintenance (and the second gene representing the cycles up to the second maintenance and so on). [58] also presents a flowchart representation of its GA methodology, which is also the base sequence of actions for all GA implementations, an cab be seen in Fig. 5.2.

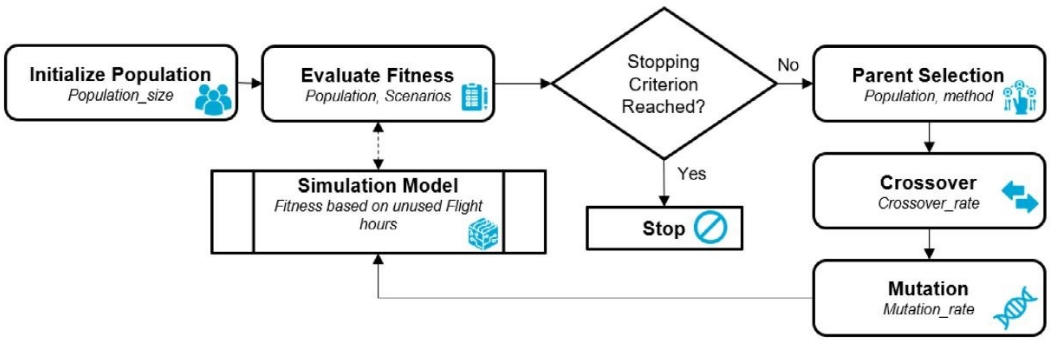


Figure 5.2: Genetic algorithm iteration process. [58]

5.6. Discussion on optimization methods

This section provides a discussion upon the presented methods to optimize the maintenance scheduling model and how they can be used in the ESV scheduling case. As part of the discussion are the main objectives of the operator in regards to the optimization, being part of the answer to the sub-question "What metrics allow for the comparison between methods?".

5.6.1. Analysis

The complexity and unpredictability of airline operations has hindered the automation and optimization of the scheduling process. As airlines might have large fleets, with different engine types and different thrust settings for each type, different operational environments and different leasing agreements for their operational and spare engines, the number of decision variables and data correlation makes the solution process computationally unpractical with classic linear programming solving methods. Furthermore, a determined optimal solution might quickly lose its feasibility or optimality as disruptions frequently occur in an airlines daily operation. Thus, the robustness of the solution might be more important to the airline as its real optimality, favoring methods that can easily cope with changes in its base scenario without the need of significant schedule changes.

As seen in 4.5, a powerful method to model these scheduling problems is using a JSSP, assigning jobs, or in this case engines, to shops, in this case an aircraft wing, storage or a maintenance workscope. As seen in [45], [21] and [58], these type of problem is typically modeled as a MILP and solved with a commercial solver. This type of problem, however, quickly reaches very high number of decision variables when evaluating long time periods, as it is the case study, favoring the use of heuristic methods to reach a good solution, although it might not be optimal, in a reasonable amount of processing time.

Dynamic programming provides one of the most common approaches to optimize the scheduling problem. Breaking down the problem in multi sub-problems might reduce the computational requirements to solve the problem, however it is an older technique that will generally be outperformed by newer techniques as genetic algorithms and reinforcement learning.

As seen, there is an abundance of research using genetic algorithms to optimize scheduling processes. This type of technique provides a very powerful tool to reach good schedules, quickly reaching a result. GAs, however, are note frequently used with JSSP because of their different structure. Furthermore it has no ”memory", that is, in case of changes in the base scenario, a new optimization has to be performed, reducing it’s robustness to operational scenarios, such as in an airline, where states frequently change.

An optimization method that seems to fit well the JSSP structure and provide a robust response is reinforcement learning. When considering maintenance and AoG costs (or loss of revenue) as constant in time for a defined engine type, finding a policy which optimizes total life cycle maintenance costs will create a system that can easily adapt to changes in the base scenario, as a good policy will be already known, even if now unfeasible, requiring less processing time to find a new feasible solution. This type of memory can be useful with the use of Deep Q-learning, as seen in [[8],[22]], in addition to the multi-agent environment, quickly reaching a new schedule when inevitable disruptions occur. Cases as the one seen in [29], being used with a multi-agent MDP, prove that it can be used for large fleets and achieve good results.

Thus, as it is possible to identify from the requirements seen for a useful tool for airlines to schedule their engine shop visits main items such as the robustness of the solution and the speed to find it, reinforcement learning seems to present a fitting optimization method. It can quickly reach new solutions once the model is trained and a good policy is determined. Furthermore, operators can also evaluate possible different scenarios, performing a manual sensitivity analysis with small computational efforts and also helping in the business decision making for the airline.

6 Conclusion

As airlines seek to reduce costs and at the same time keep and advance upon current levels of flight safety, one of the points where improvements can be made is at the scheduling of engine shop visits, as it may reduce costs with parts, spare engines, with personnel and may also increase aircraft availability, thus increasing revenue potential. As seen, ESVs can be divided into scheduled and unscheduled, being the first type related to life limited parts and continuous performance degradation, and the second part related to failing parts and airworthiness directives. Both LLPs and performance, in this case represented by the exhaust gas temperature margin, can have their threshold reaching time point estimated with linear regression based on the airline's data.

As maintenance processes progressed, older maintenance policies such as failure-based and time-based maintenance gave way to a condition-based maintenance. Although LLPs still have time-based maintenance policies, the rest of the engine system and its performance degradation are based on condition, only requiring maintenance once it is evaluated that they have failed or reached a certain threshold, or will reach it soon. This provides an additional challenge for engine maintenance scheduling optimization, as it is necessary to combine both policies in the process.

With the growth in the commercial aviation sector and the corresponding growth in airline's fleets, attention started to be redirected into the search for computational tools that could aid the maintenance departments to keep track of aircraft maintenance status and schedule their maintenance tasks in a more efficient and productive fashion. Although computational optimization methods quickly started to appear, airlines still mostly relied on decision making support simulation tools, leaving any optimization process to be manually performed by operator, as finding an optimal schedule was too computationally demanding and the frequent disruptions in an airline's schedule would quickly turn optimal solutions unfeasible.

As technology developed, new modeling and optimization techniques allow for airlines to schedule their ESVs faster and better, reaching closer to optimal results by evaluating much more scenarios than a human can. To do so, modeling mainly focused on mathematically modeling the problem with a system of equations that can be solved with a defined optimal result, modeling it as an objective function and a set of constraints in a MILP framework or with a Markov Decision Process, with most research using the latter. MILP and MDPs may have a unique optimal solution, but reaching this solution for large fleets and time scales might be computationally impractical, requiring heuristic optimization approaches to find optimal or close to optimal solutions in a quick manner.

Among the modeling and optimization methods evaluated, no research presents a comprehensive tools to optimize an ESV scheduling process considering a large fleet and all operationally significant factors, as the ones previously stated. There is thus a research gap with the opportunity to use modern modeling and optimization techniques to provide an robust and efficient tool for operators to reduce engine maintenance costs along an engine's life cycle. Among the research presented, the use of the Flexible Job-Shop Scheduling Problem model with a reinforcement learning optimization seem to be a compelling method to tackle this specific problem, providing a flexible base to apply all operational constraints and find a policy that can quickly react to disruptions and find new scheduling solutions, also allowing for airlines to test multiple scenarios and evaluate business decisions, as for example the impact of extending a leasing contract or not, considering maintenance costs.

All in all, it is possible to see that the engine shop visit scheduling process has received limited attention by research, and although a lot of operations research is done in the area of scheduling optimization, it is mostly not adapted to fit the specifications of the case. It is also clear that there is potential for improvement in an airline's maintenance cost by using modern techniques to perform this scheduling optimization, providing a tool useful for the maintenance department.

Bibliography

[1] Leveraging information technology for optimal aircraft maintenance, repair and overhaul (mro). pages xi–xv. Woodhead Publishing, 2012. ISBN 978-1-84569-982-6. doi: https://doi.org/10.1016/B978-1-84569-982-6.50012-1. URL https://www.sciencedirect.com/science/article/pii/B9781845699826500121.

[2] Dec 2023. URL https://www.iata.org/en/iata-repository/publications/economic-reports/global-outlook-for-air-transport---december-2023---report/.

[3] Shannon Ackert. Keeping score: Analysis of an engines shop visit rate, Oct 2015. URL http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/keeping_score_analysis_of_an_engine%E2%80%99s_svr____v1.pdf.

[4] Federal Aviation Administration. Ac 121-22c - maintenance review boards, maintenance type boards, and oem/tch recommended maintenance procedures, Aug 2012. URL https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1020291.

[5] Airbus. Fatal accidents. URL https://accidentstats.airbus.com/statistics/fatal-accidents.

[6] N. Aissani, B. Beldjilali, and D. Trentesaux. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Engineering Applications of Artificial Intelligence*, 22(7):1089–1103, 2009. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2009.01.014. URL https://www.sciencedirect.com/science/article/pii/S0952197609000384. Distributed Control of Production Systems.

[7] Torgny Almgren, Niclas Andréasson, Michael Patriksson, Ann-Brith Strömberg, Adam Wojciechowski, and Magnus Önnheim. The opportunistic replacement problem: theoretical analyses and numerical tests. *Mathematical Methods of Operations Research*, 76(3):289319, Aug 2012. doi: https://doi.org/10.1007/s00186-012-0400-y.

[8] Pedro Andrade, Catarina Silva, Bernardete Ribeiro, and Bruno F. Santos. Aircraft maintenance check scheduling using reinforcement learning. *Aerospace*, 8(4), 2021. ISSN 2226-4310. doi: 10.3390/aerospace8040113. URL https://www.mdpi.com/2226-4310/8/4/113.

[9] National Archives and Records Administration. ğ 33.70 engine life-limited parts., Sep 2007. URL https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-33/subpart-E/section-33.70.

[10] Dilip R. Ballal and Joseph Zelina. Progress in aeroengine technology (1939–2003). *Journal of Aircraft*, 41 (1):43–50, 2004. doi: 10.2514/1.562. URL https://doi.org/10.2514/1.562.

[11] Richard Barlow and Larry Hunter. Optimum preventive maintenance policies. *Operations Research*, 8 (1):90–100, 1960. doi: 10.1287/opre.8.1.90. URL https://doi.org/10.1287/opre.8.1.90.

[12] David Lapesa Barrera. *Aircraft Maintenance Programs*. Springer, S.L., 2022. ISBN 9783030902629.

[13] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 2021. ISBN 9781400835386.

[14] N. J. Boere. Air canada saves with aircraft maintenance scheduling. *Interfaces*, 7(3):1–13, 1977. doi: 10.1287/inte.7.3.1. URL https://doi.org/10.1287/inte.7.3.1.

[15] Richard J Boucherie, Van Dijk, and Springerlink (Online Service. *Markov Decision Processes in Practice*. Springer International Publishing, Cham, 2017. ISBN 9783319477664.

[16] Raviteja Buddala and Siba Sankar Mahapatra. Two-stage teaching-learning-based optimization method for flexible job-shop scheduling under machine breakdown. *The International Journal of Advanced Manufacturing Technology*, 100(5-8):14191432, Oct 2018. doi: https://doi.org/10.1007/s00170-018-2805-0.

[17] Eunshin Byon and Yu Ding. Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process. *IEEE Transactions on Power Systems*, 25(4):1823–1834, 2010. doi: 10.1109/TPWRS.2010.2043269.

[18] N.C. Caballé, I.T. Castro, C.J. Pérez, and J.M. Lanza-Gutiérrez. A condition-based maintenance of a dependent degradation-threshold-shock model in a system with multiple degradation processes. *Reliability Engineering System Safety*, 134:98–109, 2015. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2014.09.024. URL https://www.sciencedirect.com/science/article/pii/S0951832014002403.

[19] Jing Cai, Xin Li, and Xi Chen. Optimization of Aeroengine Shop Visit Decisions Based on Remaining Useful Life and Stochastic Repair Time. *Mathematical Problems in Engineering*, 2016:1–11, December 2016. doi: 10.1155/2016/5648360. URL https://ideas.repec.org/a/hin/jnlmpe/5648360.html.

[20] Fatih Camci. System maintenance scheduling with prognostics information using genetic algorithm. *IEEE Transactions on Reliability*, 58(3):539–552, 2009. doi: 10.1109/TR.2009.2026818.

[21] Jorge Cerdeira, Cristina Lopes, and Eliana Costa E Silva. Scheduling the repairment of aircrafts' engines. pages 259–267, 05 2017. doi: 10.1109/ICCAIRO.2017.56.

[22] Feng Cheng, Yifeng Huang, Bhavana Tanpure, Pawan Sawalani, Long Cheng, and Cong Liu. Cost-aware job scheduling for cloud instances using deep reinforcement learning. *Cluster Computing*, 25(1):619–631, Feb 2022. ISSN 1573-7543. doi: 10.1007/s10586-021-03436-8. URL https://doi.org/10.1007/s10586-021-03436-8.

[23] Thomas H Cormen, Charles Eric Leiserson, Ronald L Rivest, Clifford Stein, and Et Al. *Introduction to algorithms*. MIT Press, 3 edition, 2009. ISBN 9788120340077.

[24] Tapas K. Das, Abhijit Gosavi, Sridhar Mahadevan, and Nicholas Marchalleck. Solving semi-markov decision problems using average reward reinforcement learning. *Management Science*, 45(4):560–574, 1999. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2634824.

[25] Qichen Deng, Bruno F. Santos, and Richard Curran. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281(2):256–273, 2020. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2019.08.025. URL https://www.sciencedirect.com/science/article/pii/S0377221719306782.

[26] J Desantis. Engines turn or passengers swim: A case study of how etops engines turn or passengers swim: A case study of how etops improved safety and economics in aviation improved safety and economics in aviation. *Journal of Air Law and Commerce Journal of Air Law and Commerce*, 77, 2013. URL https://scholar.smu.edu/cgi/viewcontent.cgi?article=1305&context=jalc.

[27] EASA. Cs-25 amendment 19, May 2017. URL https://www.easa.europa.eu/en/document-library/certification-specifications/cs-25-amendment-19.

[28] Yi Feng, Lu Zhang, Zhile Yang, Yuanjun Guo, and Dongsheng Yang. Flexible job shop scheduling based on deep reinforcement learning. In *2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*, pages 660–666, 2021. doi: 10.1109/ACAIT53529.2021.9731322.

[29] Thomas Gabel and Martin Riedmiller. On a successful application of multi-agent reinforcement learning to operations research benchmarks. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 68–75, 2007. doi: 10.1109/ADPRL.2007.368171.

[30] David E Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, Reading, Mass., 1989. ISBN 9780201157673.

[31] Frederick S Hillier. *Introduction to operations research*. Mcgraw-Hill Education, Dubuque, 11 edition, 2021. ISBN 9781259872990.

[32] Honeywell. Alf502 lf507 engines definitions. URL https://havrel.honeywell.com/docs/index.cfm?content=help/engines-ALF-LF_definitions.cfm.

[33] C.-L Hsu, Tung Wan, and S.-H Chen. The study of optimum engine shop visit scheduling via genetic algorithm method. 42:141–148, 06 2010.

[34] Qiying Hu and Wuyi Yue. *Markov Decision Processes with Their Applications*. Springer Science Business Media, Sep 2007. ISBN 9780387369518.

[35] IATA. *Airline Maintenance Cost Executive Commentary FY2021*. Dec 2022. URL https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2021-mctg-report_public.pdf.

[36] Dale Weldeau Jorgenson and Roy Radner. *Optimal Replacement and Inspection of Stochastically Failing Equipment*. RAND Corporation, Santa Monica, CA, 1960.

[37] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Int. Res.*, 4(1):237285, may 1996. ISSN 1076-9757.

[38] Harry A Kinnison and Tariq Siddiqui. *Aviation maintenance management*. Mcgraw-Hill, New York, 2013. ISBN 9780071805025.

[39] Mark P. Kleeman and Gary B. Lamont. Solving the aircraft engine maintenance scheduling problem using a multi-objective evolutionary algorithm. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 782–796, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31880-4.

[40] Oliver Kramer. *Genetic Algorithm Essentials*. Cham Springer International Publishing, 2017. ISBN 978-3-319-52155-8.

[41] U Dinesh Kumar, John Crocker, J Knezevic, M El-Haram, and Springerlink (Online Service. *Reliability, Maintenance and Logistic Support : - A Life Cycle Approach*. Springer Us, New York, Ny, 2000. ISBN 9781461546559.

[42] Carlos Lagos, Felipe Delgado, and Mathias A. Klapp. Dynamic optimization for airline maintenance operations. *Transportation Science*, 54(4):998–1015, 2020. doi: 10.1287/trsc.2020.0984. URL https://doi.org/10.1287/trsc.2020.0984.

[43] Zhen Li, Shisheng Zhong, and Lin Lin. An aero-engine life-cycle maintenance policy optimization algorithm: Reinforcement learning approach. *Chinese Journal of Aeronautics*, 32(9):2133–2150, 2019. ISSN 1000-9361. doi: https://doi.org/10.1016/j.cja.2019.07.003. URL https://www.sciencedirect.com/science/article/pii/S1000936119302705.

[44] Rui Liu, Xinfeng Chen, Haijun Li, Zhao, and Zhaodong Huang. Risk sources analysis method on aero engine ifsd based on big data. In *2015 First International Conference on Reliability Systems Engineering (ICRSE)*, pages 1–6, 2015. doi: 10.1109/ICRSE.2015.7366431.

[45] Alan S. Manne. On the job-shop scheduling problem. *Operations Research*, 8(2):219–223, 1960. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/167204.

[46] M Marseguerra and E Zio. Optimizing maintenance and repair policies via a combination of genetic algorithms and monte carlo simulation. *Reliability Engineering System Safety*, 68(1):69–83, 2000. ISSN 0951-8320. doi: https://doi.org/10.1016/S0951-8320(00)00007-7. URL https://www.sciencedirect.com/science/article/pii/S0951832000000077.

[47] Yailen Martínez, Ann Nowé, Juliett Suárez, and Rafael Bello. A reinforcement learning approach for the flexible job shop scheduling problem. In Carlos A. Coello Coello, editor, *Learning and Intelligent Optimization*, pages 253–262, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-25566-3.

[48] Angel Figueroa Mayordomo, Adel Ghobbar, Stefaan Ghijs, and Eric Cator. *Engine Management: A Decision Support Tool for Strategic Engine Maintenance Planning*. doi: 10.2514/6.2010-9146. URL https://arc.aiaa.org/doi/abs/10.2514/6.2010-9146.

[49] Clinton V. Oster, John S. Strong, and C. Kurt Zorn. Analyzing aviation safety: Problems, challenges, opportunities. *Research in Transportation Economics*, 43(1):148164, Jul 2013. doi: https://doi.org/10.1016/j.retrec.2012.12.001. URL http://faculty.wcas.northwestern.edu/~ipsavage/104-13.pdf.

[50] Yosef S Sherif. Optimal maintenance schedules of systems subject to stochastic failure. *Microelectronics Reliability*, 22(1):15–29, 1982. ISSN 0026-2714. doi: https://doi.org/10.1016/0026-2714(82)90047-6. URL https://www.sciencedirect.com/science/article/pii/0026271482900476.

[51] Gerardo I Simari and Simon D Parsons. *Markov Decision Processes and the Belief-Desire-Intention Model*. Springer Science Business Media, Sep 2011. ISBN 9781461414728.

[52] Allianz Global Corporate Specialty. *Global Aviation Safety Study: A review of 60 years of improvement in aviation safety*. Dec 2014. URL https://www.agcs.allianz.com/content/dam/onemarketing/agcs/agcs/reports/AGCS-Global-Aviation-Safety-2014-report.pdf.

[53] Armin Stranjak, Partha Dutta, Mark Ebden, Alex Rogers, and Perukrishnen Vytelingum. A multi-agent simulation system for prediction and scheduling of aero engine overhaul. pages 81–88, 05 2008. doi: 10.1145/1402795.1402811.

[54] Jianzhong Sun, Yuanchu Sun, Dan Chen, and He Liu. An integrated life-cycle maintenance modeling and simulation framework for aero engine fleet management decision support. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 0(0):1748006X221099767, 0. doi: 10.1177/1748006X221099767. URL https://doi.org/10.1177/1748006X221099767.

[55] Richard S Sutton and Andrew Barto. *Reinforcement learning : an introduction*. The Mit Press, Cambridge, Ma ; Lodon, 1998. ISBN 9780262039246.

[56] Yuo-Tern Tsai, Kuo-Shong Wang, and Hwei-Yuan Teng. Optimizing preventive maintenance for mechanical components using genetic algorithms. *Reliability Engineering System Safety*, 74(1):89–97, 2001. ISSN 0951-8320. doi: https://doi.org/10.1016/S0951-8320(01)00065-5. URL https://www.sciencedirect.com/science/article/pii/S0951832001000655.

[57] Jorne Van den Bergh, Philippe De Bruecker, Jeroen Belien, and Jonas Peeters. *Aircraft maintenance operations: state of the art*. HUBrussel; Brussel, 2013-11-01.

[58] Tim van der Weide, Qichen Deng, and Bruno F. Santos. Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers Operations Research*, 141:105667, 2022. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2021.105667. URL https://www.sciencedirect.com/science/article/pii/S0305054821003671.

[59] Andrija Volkanovski, Borut Mavko, Tome Boevski, Anton auevski, and Marko epin. Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system. *Reliability Engineering System Safety*, 93(6):779–789, 2008. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2007.03.027. URL https://www.sciencedirect.com/science/article/pii/S0951832007001196.

[60] Shevantha K. Weerasekera. SAE International, 2020. ISBN 978-0-7680-9971-3. URL https://app.knovel.com/hotlink/toc/id:kpIMROAEC1/introduction-maintenance/introduction-maintenance.

[61] Zimin (Max) Yang, Dragan Djurdjanovic, and Jun Ni. Maintenance scheduling in manufacturing systems based on predicted machine degradation. *Journal of Intelligent Manufacturing*, 19(1):8798, Jul 2007. doi: https://doi.org/10.1007/s10845-007-0047-3.

[62] Xiaojun Zhou, Lifeng Xi, and Jay Lee. Opportunistic preventive maintenance scheduling for a multi-unit series system based on dynamic programming. *International Journal of Production Economics*, 118(2):361–366, 2009. ISSN 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2008.09.012. URL https://www.sciencedirect.com/science/article/pii/S0925527308003472.