Condition-Based Maintenance scheduling of an aircraft fleet under partial observability
A Deep Reinforcement Learning approach

Tseremoglou, Iordanis; Santos, Bruno F.

DOI
10.1016/j.ress.2023.109582

Publication date
2024

Document Version
Final published version

Published in
Reliability Engineering and System Safety

Citation (APA)
Tseremoglou, I., & Santos, B. F. (2024). Condition-Based Maintenance scheduling of an aircraft fleet under partial observability: A Deep Reinforcement Learning approach. *Reliability Engineering and System Safety*, *241*, Article 109582. https://doi.org/10.1016/j.ress.2023.109582

Reliability Engineering and System Safety 241 (2024) 109582

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Condition-Based Maintenance scheduling of an aircraft fleet under partial observability: A Deep Reinforcement Learning approach

Iordanis Tseremoglou *, Bruno F. Santos

Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology, HS2629 Delft, The Netherlands

ARTICLE INFO

Keywords:
Condition-Based Maintenance (CBM)
Partially Observable Markov Decision Process (POMDP)
Partially Observable Monte–Carlo Planning (POMCP)
Prognostics
Planning under uncertainty
Deep Reinforcement Learning

ABSTRACT

In the Condition-Based Maintenance (CBM) context, the definition of optimal maintenance plans for an aircraft fleet depends on an efficient integration of : (i) the probabilistic predictions of the health condition of the components and (ii) the stochastic arrival of the corrective maintenance tasks, together with consideration of the preventive maintenance tasks as defined in the Maintenance Planning Document (MPD) . To this end, in this paper, we present a two-stage dynamic scheduling framework to solve the aircraft fleet maintenance scheduling problem under a CBM strategy in a disruptive environment. In the first stage of the framework, we address the uncertainty in the predicted health state of the monitored components by planning the optimal maintenance policy based upon the *belief* state-space of the health of the components. The decision-making process is formulated as a Partially Observable Markov Decision Process (POMDP) and is solved using the Partially Observable Monte Carlo Planning (POMCP) algorithm, considering the aircraft maintenance scheduling problem requirements. In the second stage, a Deep Q-Network (DQN) is developed, that integrates the defined maintenance policy of the monitored components within the scheduling of the aircraft fleet's preventive and corrective maintenance tasks. Our model, through a rolling horizon approach, continuously creates and adjusts the maintenance schedule, reacting to new updated task information, where the availability of maintenance resources constraints the execution of each task. The proposed framework was tested on a case study from a large airline and the performance was evaluated against the current state practice of the airline. The results show that our model can schedule 96.4% of monitored components on-time. As a consequence of this, a 46.2% maintenance cost reduction is achieved for the considered monitored components relative to a corrective maintenance approach.

1. Introduction

Maintenance, Repair and Overhaul(MRO) activities represent around 10%–15% of an airline's operational costs, while at the same time they account for 80% of the ground time [1]. Hence, optimization of the maintenance schedule is of high interest both for the scientific community and the aviation industry.

Nowadays, aircraft maintenance is either following the preventive or the corrective approach. The preventive approach is the most frequently applied methodology in the aviation and imposes maintenance interventions on fixed intervals, e.g., Flight Hours (FHs), Flight Cycles (FCs) or Calendar Days (DYs). These intervals do not take the current health status of the components into consideration. Thus, parts may be replaced without necessity leading to waste of resources (labor hours/spare parts) and improved operational costs. This strategy is implemented through the preventive maintenance tasks provided in the Maintenance Planning Document (MPD) and included in scheduled maintenance checks, also referred as letter checks.

All the other tasks not falling under the category of letter checks are referred as corrective maintenance tasks. According to the corrective strategy, a system is replaced/repaired only when it fails, thus the lifetime of the component is fully exploited. However, the stochastic nature of the corrective maintenance tasks creates disruptions to the maintenance schedule, inducing high related maintenance costs.

In order to overcome the limitations of the former strategies, aircraft maintenance providers are shifting towards a Condition-Based Maintenance (CBM) logic. This is reflected in the constantly growing number of condition-monitoring technologies, which are mostly based on automatic sensor-based collection data, that have been developed over the years for different systems of the aircraft (hydraulic systems, engines, structures). Using these technologies, the CBM approach aims to provide the maintenance planner with a constant insight into the health state of the monitored system and, subsequently, project failure events, hence decreasing the amount of unnecessary maintenance actions and at the same time, avoiding unforeseen failures.

* Corresponding author.
E-mail address: I.Tseremoglou@tudelft.nl (I. Tseremoglou).

https://doi.org/10.1016/j.ress.2023.109582
Received 9 January 2023; Received in revised form 1 June 2023; Accepted 15 August 2023
Available online 25 August 2023

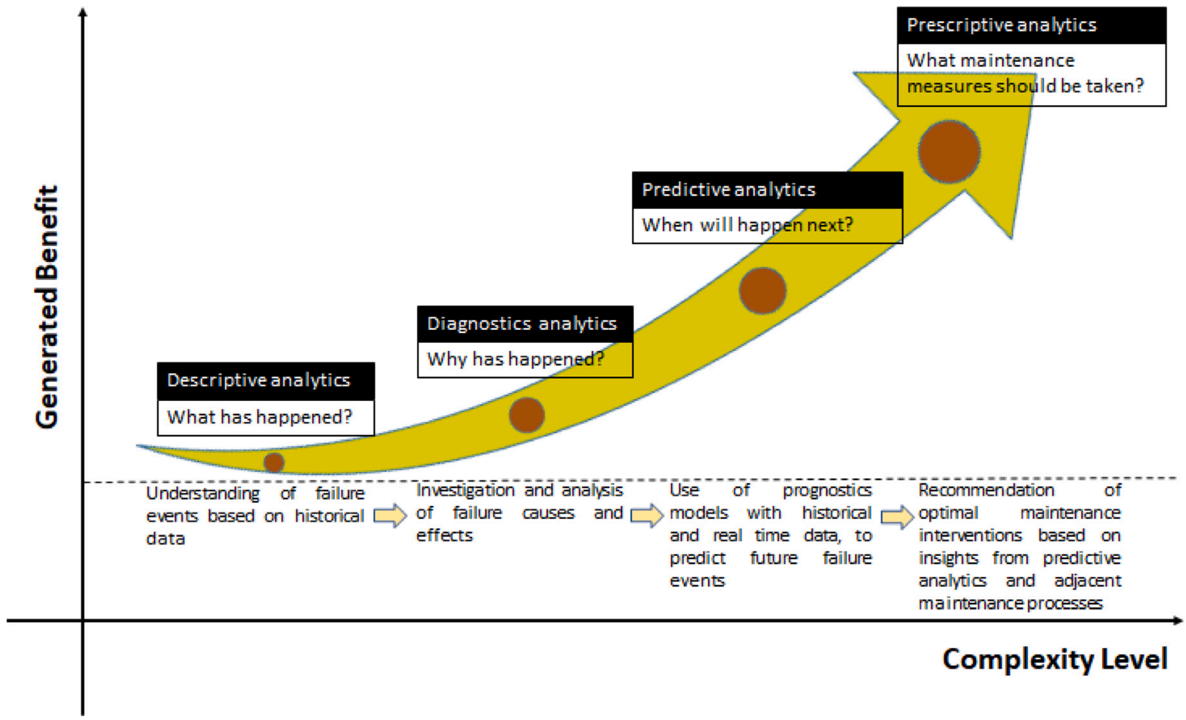

Fig. 1. Evolution of CBM with respect to data analytics level (based in [3]).

In the CBM context, different levels of data analytics can be employed: descriptive, diagnostic, predictive, and prescriptive [2]. Inspired by the analysis performed in [3], the evolution of the CBM approach with respect to the analytics level is summarized in Fig. 1.

Prescriptive analytics offers the highest degree of complexity and maturity within the CBM context. They utilize the knowledge (usually in the format of components' Remaining Useful Life (RUL) predictions, i.e., the estimated amount of usage that the component has until it becomes non-operational) from predictive analytics and integrate it with other information such:

- required resources (spare parts, workhours, machinery) and their availability
- required ground time for the execution of the corresponding maintenance task
- available maintenance opportunities

to derive the maintenance schedule that optimizes a target function, which is shaped by the planning objectives of the maintenance planner. In this paper, we use the term CBM to capture this last and prescriptive step of scheduling maintenance activities denominated by prescriptive analytics.

The conventional aircraft maintenance scheduling problem, which refers to the optimal allocation of the preventive and corrective maintenance tasks to the best maintenance opportunities, is a very challenging problem because of its combinatorial nature. However, scheduling within the CBM context becomes even more challenging for the following reasons:

1. First and foremost, there is a specific degree of uncertainty included in the RUL predictions of the prognostics algorithms. The range in the predicted RULs creates great ambiguity for the maintenance planner with respect to when to schedule the component for maintenance.
2. Secondly, it will not be possible nor relevant to perform maintenance in all aircraft components according to a CBM approach. Therefore the scheduling of the prognostics-driven tasks has to be combined with the scheduling of the existing preventive and corrective maintenance tasks.
3. Following this, the continuous update of the RUL predictions together with the unexpected arrival of the corrective maintenance tasks may create disruptions that compromise the feasibility and the efficiency of the previously generated maintenance schedule. For this reason, in a practical context, the maintenance schedule needs to be continuously adjusted, whereas at the same time, the operational availability of the fleet must be ensured and last-minute schedule changes must be prevented.

As it has become evident, it is very difficult for a human maintenance planner to consider and translate all these different types of inputs and constraints into an optimal maintenance schedule for the aircraft fleet. Therefore, this research paper aims to build a prescriptive two-stage maintenance scheduling framework that addresses the former challenges in order to support decision-making for the maintenance planner in a "hybrid" CBM context, where each aircraft is having multiple components that are maintained through the preventive, the corrective or the prognostics-driven approach, resulting to a mixture of preventive, corrective or prognostics-driven maintenance tasks. The maintenance date for the relevant components is determined by allocating an aircraft to a maintenance slot, as determined by the output of the framework. Moreover, since the model is supposed to serve as a decision tool in a quasi-real time environment, it should be computationally efficient. It is noted that although the model is shown here in an airline environment, we believe that its applicability extends to other fields of maintenance planning, like the rail or the maritime industry.

In the first stage of scheduling, we acknowledge and address the imprecision and uncertainty from prognostics, by adopting a Partially Observable Markov Decision Process (POMDP) framework. Contrary to Markov Decision Processes (MPDs), where the true state of the equipment is known with certainty and the decision-maker chooses an action based on that state, in POMDPs the maintenance policies are conditions on *beliefs* over the state of the equipment. The POMDP is solved using the Partially Observable Monte Carlo Planning algorithm, developed by Silver and Veness [4], favorably tailored to the requirements of the aircraft maintenance scheduling problem.

In the second stage, we utilize a Deep Reinforcement Learning (DRL) algorithm to find the most suitable maintenance opportunity in the airline schedule for the allocation of all the different types of tasks (prognostics-driven, preventive, corrective). A Deep Q Network is developed that considers a hierarchy of scheduling objectives and is constrained by the availability of material, machinery, method, and manpower (4M). The final output of the framework is a dynamic schedule for the prognostics-driven, preventive and corrective maintenance tasks of each aircraft within the fleet.

To illustrate our approach a real case study, with a fleet of 34 wide-body aircraft, each equipped with multiple components that are maintained through the preventive, corrective or the prognostics-driven approach, is considered. The evaluation includes a comparison of the efficiency of the proposed framework against the actual airline maintenance schedule, a sensitivity analysis of the effect of uncertainty on the number of tasks going due and the last-minute schedule changes, and an assessment of cost reduction due to the introduction of prognostics-driven tasks.

The remainder of this paper is organized as follows: In Section 2, the relevant literature pertaining to fleet maintenance scheduling problem with partial information is presented along with the identified research contribution. The aircraft maintenance requirements, objectives, and the general problem formulation are described in Section 3. The POMDP problem formulation and the related POMCP based-scheduling algorithm are discussed in detail in Section 4. The DRL scheduling model for the aircraft fleet is described in Section 5. In Section 6, a case study is performed for a fleet of 34 wide-body aircraft, each of which has a list of open maintenance tasks that are updated on a continuous basis. Finally, Section 7 summarizes the research with concluding remarks and recommendations for future work.

2. Literature review

Maintenance optimization for a fleet of assets is a challenging problem to solve. Moreover, it is inevitably faced with multiple constraints such as the manpower, material, machinery, and maintenance slot capacity and availability. During the recent years, maintenance has been studied from various perspectives [5]. As discussed previously, we distinguish two broader categories of approaches pertaining to maintenance scheduling: the condition-based, and the "traditional" preventive and corrective approach. Individually they are widely discussed in the literature.

Focusing on maintenance optimization problems of an aircraft fleet considering the preventive and/or the corrective maintenance approach, Deng et al. [6] proposed a Dynamic Programming (DP) approach that optimizes the A and C-checks schedule for a heterogeneous fleet of 40 aircraft for a period of four years. Their objective is to minimize the wasted interval between checks. The same case study with a similar objective was studied by Andrande et al. [7], who developed a DRL algorithm to schedule letter checks. Lagos et al. [8] formulated the combination of aircraft maintenance scheduling and tail assignment problem as an MDP. The problem is solved using Approximate Dynamic Programming (ADP), where the estimation of future costs is provided by means of rolling horizon techniques and value function approximation. Most recently, van Kessel et al. [9] developed a MILP framework for the dynamic scheduling of maintenance tasks in a disruptive environment, where the execution of each task is constrained by the availability of resources.

However, during the recent years CBM has gained increased attention as a preferred alternative approach to preventive and corrective maintenance. As such it has been studied for various assets of various degradation models [10]. The derived maintenance policy is based on the knowledge about the system state. Observation data, such as sensor measurements or RUL predictions provide information about the state of the system. However, in the vast majority of cases, observation data provide only partial information about the system state. For this reason, maintenance policy optimization for a system under partial observability as formulated as a Partially Observable Markov Decision Process (POMDP). We refer the interested readers to the book by Powell and Ryzhov [11] for a comprehensive analysis of different decision-making methods under partial information.

Among some indicative examples, Nguyen et al. [12] developed a dynamic condition-based maintenance and inspection framework using a POMDP model for a system subject to continuous degradation and imperfect inspections. Liu et al. [13] addressed a multi-type inspection and online monitoring problem for gas turbines within a POMDP framework, and they solved it using a combination of the value iteration technique and the λ-minimization algorithm. Song et al. [14] integrated Value of Information analysis within a POMDP framework that uses multiple transition models for different deterioration rates to derive the optimal maintenance policy for a corroding beam. Zhao and Smidts [15] proposed a reinforcement learning approach consisting of a learning and a planning component to improve the knowledge of system degradation and compute the optimal maintenance policy respectively. Most of the existing studies correspond to low dimensional domains as they are calculating the optimal maintenance policy for a single-component system. What is more, in all but a few studies [15], it is assumed that the decision-maker has knowledge of the parameters of the system degradation model.

However, in the aircraft maintenance scheduling problem, the number of states and actions can scale exponentially depending on the number of considered aircraft and the different types of tasks, which could take significant computational time and memory usage when solved by any conventional solution scheme. This relates to the so-called *curse of dimensionality*. An approach towards addressing the curse of dimensionality as well as the less-known curse of history, where the number of belief-contingent maintenance plans grows exponentially with the planning horizon is proposed by Silver and Veness [4]. They develop a Partially Observable Monte Carlo Planning (POMCP) algorithm for online planning for large POMDPs. Papakonstantinou and Shinozuka [16] resort to a point-based value iteration solver to derive the optimal maintenance policy for a concrete structure of a considerably large state-space of 332 states.

Furthermore, recent studies have proposed the use of DRL for maintenance planning within a high-dimensional CBM context, as it has showcased unparalleled ability of learning and solving high-dimensional and complex environments which are described by continuous state features, in a computationally efficient manner. Andriotis and Papakonstantinou [17] proposed a Deep Centralized Multi-Agent Actor-Critic model to solve POMDPs for optimal decision-making in complex, non-stationary, partially observable engineering environments with large state and action spaces. Building on this, Andriotis and Papakonstantinou [18], developed a multi-agent DRL framework to derive the optimal maintenance policy for multiple components having a degradation represented by 4 states. Zhang and Si [19] proposed a customized DRL model to optimize the maintenance of multi-component systems having a degradation that follows a compound Poisson and Gamma process. Mohammadi and He [20] applied a DRL-based approach on maintenance and renewal planning of railways, where they consider both preventive and condition-based maintenance, along with budget constraints. However, in an airline environment, the maintenance planner is faced with more complex operational constraints and also with the frequent arrival of multiple unexpected corrective maintenance tasks that, if they are not timely and efficiently planned, might create disruptions to the busy flight schedule.

Focusing on CBM planning on the commercial aviation sector, De Pater and Mitici [21] formulated a rolling horizon maintenance planning approach for multiple multi-component systems. Their model integrates the RUL prognostics with the management of a limited stock of repairable components, while also considering the availability of maintenance slots. Their approach was illustrated on a fleet of 13 aircraft, each equipped with a Cooling System consisting of four Cooling Units. In a similar fashion, De Pater et al. [22] proposed an alarm-based dynamic maintenance framework for a fleet of 20 aircraft,

Table 1
Defined sets for the two-stage scheduling model.

Set	Description
$u \in U$	Set of monitored components
$g \in G$	Set of open maintenance tasks $G = G_{prev} \cup G_{corr} \cup G_{progn}$
$g \in G_{prev}$	Subset of preventive maintenance tasks
$g \in G_{corr}$	Subset of corrective maintenance tasks
$g \in G_{progn}$	Subset of prognostics-driven maintenance tasks corresponding to monitored components U
$r \in R$	Set of aircraft registrations
$u \in U_r$	Subset of monitored components belonging in aircraft registration r ($U_r \subset U$)
$m \in M$	Set of maintenance slots
$m \in M_{Fixed}$	Subset of fixed maintenance slots
$m \in M_{Flexible}$	Subset of flexible maintenance slots
$w \in W$	Set of workforce skills

each equipped with 2 engines. Lee and Mitici [23] addressed the predictive maintenance planning problem of turbofan engines using a DRL approach. However, all the previous works focus only on maintenance planning of the monitored components (CBM tasks) without considering the maintenance planning for the rest of the aircraft components that are still addressed through the preventive and corrective maintenance approach. Moreover, including in a CBM planning context the full spectrum of constraints and planning objectives that are encountered in a real commercial airline environment has not been studied yet.

To sum up, maintenance optimization for a fleet of assets has been studied from several angles. However, a dynamic framework, jointly optimizing the allocation of preventive and corrective maintenance tasks from multiple aircraft while considering the uncertainty of RUL predictions driving the allocation of the CBM tasks under real 4M constraints, has not been reported to the literature up to date. More specifically, the main contributions of our research are summarized in the following:

1. For the first time, a novel dynamic scheduling framework that considers all the different types of tasks (preventive, corrective and prognostics-driven) and 4M constraints encountered in a real CBM airline environment, is developed. The proposed framework performs well according to multiple Key Performance Indicators (KPIs) and, more importantly, is computationally efficient and fast for real-time implementation in an airline environment.
2. Second, an innovative approach, for deriving the optimal maintenance policy for monitored components having RUL predictions with uncertainty, is proposed, based on a modified version of the POMCP algorithm. It is also considered that the decision-maker has no knowledge over the components' degradation model. At each time step of the decision-making, the algorithm uses real-time RUL predictions to improve the decision-makers' knowledge of the degradation process.
3. Third, a novel Deep Reinforcement Learning (DRL) approach, with a set of elaborately-designed state features capturing the planning objectives and the full spectrum of 4M constraints of the aircraft fleet maintenance scheduling problem, is developed.

Table 2
Defined parameters for the two-stage scheduling model.

Parameter	Unit	Description
t	Date	Current date of scheduling
δ_r	FCs	Average flight daily usage of aircraft r, $r \in R$
CP	Days	Schedule change prevention days
Due_g	Date	Due date of task g, $g \in G_{prev} \cup G_{corr}$
GR_m^w	Hours	Amount of available labor hours of skill type w f maintenance slot m
GR_g^w	Hours	Amount of required labor hours of skill type ts to perform task g
$Duration_m$	Hours	Duration of maintenance slot s
$Duration_g$	Hours	Duration of maintenance task g
$Arrival_g$	Date	Creation date of maintenance task g
$C_{u_r}^{corr}$	Euros	Corrective maintenance cost for component u installed in aircraft r
$C_{u_r}^{prev}$	Euros	Preventive maintenance cost for component u installed in aircraft r
$DD_{g,m}$	[–]	1 if the start date of maintenance slot m is before the due date of task g
$Material_{g,m}$	[–]	1 if the required material for the execution of maintenance task g is available before the start date of maintenance slot m
$Machinery_{g,m}$	[–]	1 if the required equipment for the execution of maintenance task g is available before the start date of maintenance slot m
$ACtype_{r,m}$	[–]	1 if the aircraft type of maintenance slot m matches the type of aircraft registration r

Table 3
Defined decision variables for the two-stage scheduling model.

Decision variable	Description
$T_{g,m}$	Binary, 1 if task g is assigned to slot m
$AC_{r,m}$	Binary, 1 if aircraft r is assigned to slot m

3. Problem formulation

3.1. Problem definition and scope

The problem we are addressing can be summarized as follows: Let us consider an aircraft fleet, where each aircraft is having multiple components. Some of these components are maintained according to the preventive or the corrective approach — generating the corresponding preventive and corrective maintenance tasks.

Moreover, some aircraft from the fleet are having components that are monitored through sensors. We assume that the monitored components of these aircraft are subject to deterioration according to a continuous-time discrete-state Markov chain. However, there is no specific degradation model that describes the deterioration of these components. Instead, for every monitored component, after every flight, a prognostics model produces a RUL prediction that follows the normal distribution $\sim N(\mu, \sigma^2)$. Based on these RUL predictions, the corresponding prognostics-driven tasks are generated.

This complete set of tasks is continuously updated, either due to irregular arrival of corrective tasks, such as faults reported by the pilots, or due to new RUL predictions obtained from the prognostics. The overarching goal is to allocate these tasks to the available maintenance opportunities, such that the airline operator's objectives and the corresponding KPIs are satisfied. According to [3], the operator's objective can be summarized in the realization of high asset availability for revenue generation through the assurance of reliable aircraft operations.

Due to the different nature and requirements of the considered tasks, we solve the aircraft fleet maintenance scheduling problem in two stages. In the first stage, we derive the optimal maintenance policy for the monitored components, by developing a POMDP formulation that captures the stochastic relation between the component's true health state and the observed RUL prediction. In the second stage, through a DRL approach, we incorporate the defined maintenance policy of the monitored components within the scheduling of the preventive and corrective maintenance tasks in order to devise a maintenance schedule for the aircraft fleet. The defined sets, parameters and decision variables are described in Tables 1–3.

3.2. Scheduling framework

The proposed two-stage scheduling framework is summarized in Fig. 2. The modeling blocks are represented with a gray background.

In the first stage, we solve the scheduling problem of the prognostics-driven tasks, i.e., we seek to define the optimal maintenance policy and corresponding date for each monitored component. Decisions about maintenance intervention are made at discrete time points of the planning horizon, referred to as decision epochs. These decision epochs could be pre-defined, e.g. according to fixed intervals of 1 day, or just triggered by the arrival of new information from the sensors. The sequence of events is as follows: At the beginning of every decision epoch, a new RUL probabilistic prediction for every monitored component is obtained from the prognostics model. Following this new prediction/observation, the belief regarding the health state of the component is updated. Based on the updated belief state, the availability of maintenance slots and resources, and the related maintenance costs, the prognostics-driven tasks scheduling algorithm determines the optimal point in time to perform maintenance. The former decision-making process is solved by a modified version of the POMCP algorithm, discussed in Section 4. The output of the algorithm is the optimal maintenance date for every monitored component.

In the second stage, we consider for scheduling the prognostics-driven tasks that are defined from the optimal maintenance dates of the monitored components, together with the current list of corrective and preventive maintenance tasks for the aircraft fleet. We use a DRL approach to solve the aircraft fleet maintenance scheduling problem. The DRL algorithm, taking into account the current maintenance schedule, the set of different types of open maintenance tasks, the available maintenance slots, and the available resources, produces an updated maintenance schedule for the aircraft fleet for a time horizon of multiple weeks. This updated maintenance schedule will form the starting solution when new information for the maintenance tasks is obtained.

3.3. Inputs of the scheduling framework

3.3.1. Inputs for the prognostics-driven tasks maintenance scheduling model

- **Predicted RUL distributions**: A distribution that captures the amount of time left until the end-of-life of the component. For every monitored component, the prognostics model produces a predicted RUL distribution, based on the information it receives from the sensors.
- **Maintenance slots**: The maintenance slots are time slots intervals which are reserved to execute maintenance. Each slot has a start date, an end date, and a designated aircraft type. Maintenance slots can be further subdivided into two categories:
  - *Fixed slots*: For the fixed slots, the assigned aircraft tail number is predefined. Fixed maintenance slots are usually scheduled several weeks in advance and consist of more extensive maintenance operations such as letter checks.
  - *Flexible slots*: For flexible maintenance slots, the aircraft registration is variable and a maintenance scheduler is free to decide which aircraft to allocate to the slot, as long as the considered aircraft type matches the slot designated aircraft type.
- **Maintenance costs**: For every monitored component, there is a related preventive and corrective maintenance cost. The corrective maintenance cost is always higher than the preventive maintenance cost.
- **Resources**: The execution of a maintenance task requires the availability of specific resources, referenced as 4M requirements. Below an explanation of the 4M requirements that are going to be considered in this research is provided:
  - *Method*: Each maintenance task requires a specific amount of ground time to be executed. This means that a maintenance task can only be allocated to maintenance slots with a duration equal to or higher than the required ground time.
  - *Machinery*: Some maintenance tasks require the availability of specific types of equipment and/or tools to be executed. For each task, the date after which the required machinery would be available, is provided.
  - *Material*: Similar to machinery, some tasks require the availability of specific parts/consumables. For each task, the date of material availability is provided.
  - *Manpower*: Each maintenance task is associated with specific manpower requirements. The manpower is divided by skill type. There is a daily workforce schedule, organized in shifts, where the available workhours per skill are described. A task can be allocated to a maintenance slot, only if the corresponding required workhours per skill of the task are satisfied, in means of the available workforce in this specific slot.

3.3.2. Inputs for the aircraft fleet maintenance scheduling model

- **Current maintenance schedule**: A feasible and updated schedule considering past information regarding the prognostics-driven, preventive and corrective tasks. The maintenance schedule details the allocation of maintenance tasks before their due date to the available maintenance slots, based on the available resources. In case a maintenance schedule does not exist, the proposed scheduling framework can be used to generate an initial maintenance schedule.
- **Maintenance slots**: An explanation of the maintenance slots is provided in Section 3.3.1.
- **Open maintenance tasks**: In the aircraft maintenance context, each aircraft has a backlog of open tasks. These tasks correspond to jobs required to be executed within a specified time interval to ensure aircraft airworthiness. Moreover, each task comes with the 4M requirements, namely the required execution time (Method), workforce (Manpower), Material, and Machinery. In order for the airline to be able to schedule a task, all 4M requirements need to be satisfied. The different types of tasks that are going to be considered in the context of this research are described below:
  - *Preventive maintenance tasks*: The preventive maintenance tasks are prescribed in the Maintenance Planning Document (MPD) provided by the aircraft manufacturer. This type of tasks is performed in fixed periodic inspection intervals that come in the form of FHs, FCs or DYs. Once the task is performed, the corresponding interval is reset.
  - *Corrective maintenance tasks*: The corrective maintenance tasks are non-scheduled tasks that can be a result of a fault reported by the pilots or of a finding during the execution of a preventive maintenance task. In case a corrective task corresponds to a component included in the Minimum

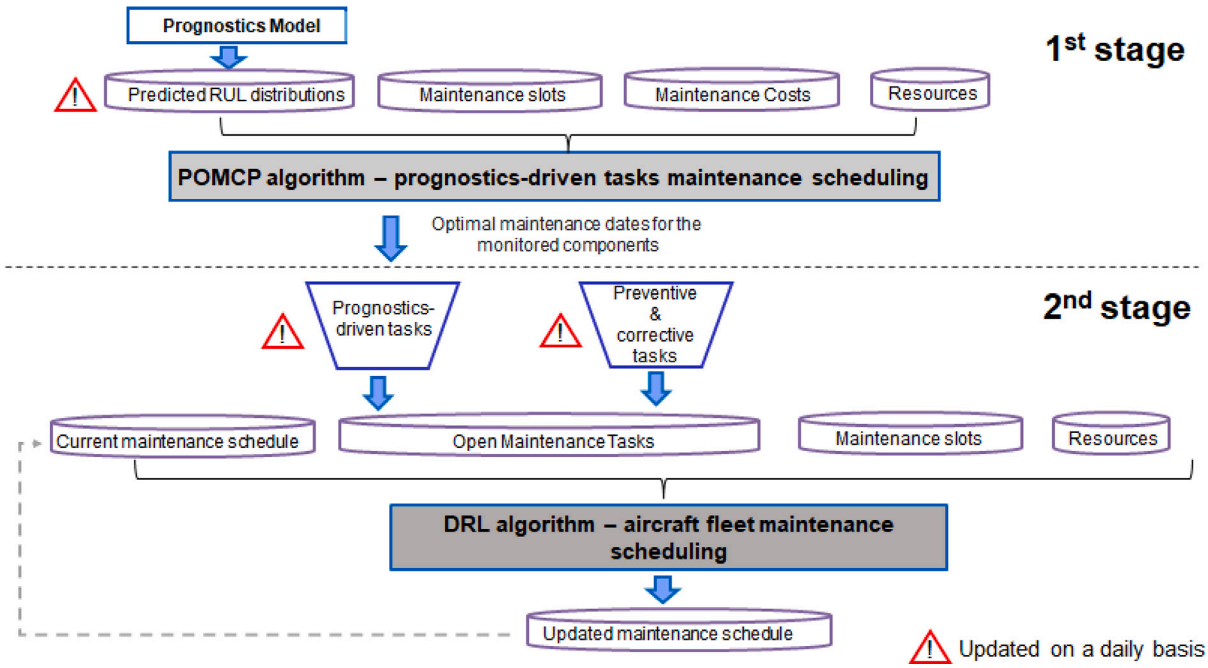

Fig. 2. Overview of the two-stage scheduling framework, with the algorithms used in both stages highlighted in gray.

Equipment List (MEL), then the aircraft is allowed to be airworthy with the corrective task unresolved, as long as the due date of the task, as specified in the MEL, is not exceeded. Moreover, a corrective maintenance task can also correspond to Non-Safety Related Equipment (NSRE), and as such the aircraft remains airworthy even after the due date of the task is exceeded. Finally, there are corrective maintenance tasks created as a result of modifications recommended by the manufacturer (Service Bulletins -SBs) or by the aviation authorities (Airworthiness Directives - ADs). The former type of tasks needs to be executed only once.

– *Prognostics-driven tasks*: The prognostics-driven tasks refer to the maintenance of systems or components that are monitored through sensors permanently. The collected information is passed to prognostics models that predict the RUL. Maintenance actions are triggered only when there is strong evidence of failure risk, hence decreasing the number of unnecessary maintenance actions and, at the same time, avoiding unforeseen failures (and corresponding unscheduled maintenance events). For the purposes of the study, the prognostics-driven tasks correspond to components that are not critical for the safe operation of the aircraft, i.e., the component can fail without the flight safety being jeopardized.

- **Resources**: An explanation of resources is provided in Section 3.3.1

3.4. Maintenance planning objectives

Within commercial airlines, maintenance planners have to generate maintenance schedules that are both efficient and stable. Increased schedule efficiency leads to increased aircraft availability for operations, which subsequently increases the revenue for the airline. Maintenance schedule stability is reflected in flight schedule stability, hence minimizing disruptions or delays in passenger itineraries. Following this mindset of a maintenance planner within a real airline environment, and the analysis performed in [9], we consider the following four planning objectives:

1. **Task execution**: The first objective is to execute tasks ahead of their due date. When the due date of a task is exceeded, the aircraft is no longer airworthy and it has to be grounded until the corresponding task is performed. This induces major costs for the airline, as the aircraft is not available for operations.
2. **Aircraft operational availability**: This is a two-fold objective: The aircraft visits to the hangar for maintenance should be minimized and, at the same time, the ground time associated with these visits when they occur should be minimized. This means that as many open tasks as possible should be addressed at the fixed maintenance slots, since during these dates the aircraft will visit the hangar for maintenance anyway, as part of a letter check. Furthermore, the remaining tasks that cannot be allocated at the fixed slots, should be assigned to as few flexible slots as possible. At the same time, these flexible slots have to be efficiently used, reducing the associated ground time wasted. For example, a task with a required duration of execution of 5 h, it is more efficient to be assigned to a maintenance slot with a duration of 6 h rather than a slot with a duration of 10 h.
3. **Schedule stability**: The third objective is to guarantee schedule stability by minimizing schedule changes due to the continuous update of the maintenance tasks. A schedule change is defined as a change in the aircraft registration assigned to a flexible maintenance slot, when compared with the existing schedule. Minimizing schedule changes contributes towards increased reliability of the established flight schedule, reducing the chance of network disruptions such as flight delays or cancelations. It is noted that reallocating tasks between maintenance slots of the same aircraft registration is not considered a schedule change, since it does not have an impact on the operational availability of the aircraft. However, in case a schedule change cannot be prevented, then the number of days of notice is important, i.e., an aircraft allocation change one day before the day of operations will have more severe effects on the flight schedule than a change 10 days ahead.
4. **Task interval utilization**: The last objective is to plan tasks at the optimal moment in time. A metric used in the airlines to quantify the efficiency of task scheduling is the task interval utilization, which can be defined as the ratio of the scheduled

day of the task to the due day of the task. The preventive maintenance tasks are executed in fixed intervals, meaning that the objective is to schedule them as close as possible to the end of the respective interval, as this would minimize the repetition of maintenance interventions in the long run. For the prognostics-driven tasks, the objective is to schedule them as close as possible to the end of the RUL of the monitored component. So, for both the preventive and the prognostics-driven maintenance tasks, the objective is to achieve a high task interval utilization. On the contrary, the corrective maintenance tasks have to be executed as soon as possible for quality reasons. So the objective for these tasks is to achieve a low task interval utilization.

3.5. Constraints

$$\sum_{r\in R} AC_{r,m} \leq 1 \qquad \forall m \in M \tag{1}$$

$$\sum_{m\in M} DD_{g,m} \cdot T_{g,m} = 1 \qquad \forall g \in G \tag{2}$$

$$\sum_{m\in M} (1 - Material_{g,m}) \cdot T_{g,m} = 0 \qquad \forall g \in G \tag{3}$$

$$\sum_{m\in M} (1 - Machinery_{g,m}) \cdot T_{g,m} = 0 \qquad \forall g \in G \tag{4}$$

$$\sum_{g\in G} GR^{w}_{g} \cdot T_{g,m} \leq GR^{w}_{m} \qquad \forall w \in W, m \in M \tag{5}$$

$$Duration_{g} \cdot T_{g,m} \leq Duration_{m} \qquad \forall m \in M, g \in G \tag{6}$$

$$\sum_{m\in M} (1 - ACtype_{r,m}) \cdot AC_{r,m} = 0 \qquad \forall r \in R \tag{7}$$

$$T_{g,m}, AC_{r,m} \qquad \in \{0,1\} \tag{8}$$

The set of constraints (1) imposes that only one aircraft can be assigned to a maintenance slot. According to the set of constraints (2), a task can only be assigned to a maintenance slot with a start date earlier than its due date. The set of constraints (3)–(6) guarantees the satisfaction of the 4M requirements when a task is allocated to a maintenance slot. The set of constraints (7) ensures that aircraft of a specific type can only be scheduled to slots of a matching aircraft type, while the set of constraints (8) defines the type of the decision variables.

4. Prognostics-driven tasks scheduling

In this section, we provide an analysis of the methodology and the algorithm used to derive the optimal maintenance policy over the intended planning horizon for the monitored components considered in the aircraft fleet.

4.1. Problem formulation

4.1.1. State, observation and action modeling

Let R denote the set of the considered aircraft registrations in the aircraft fleet. Each aircraft has a set of different and independent monitored components U_r, $r \in R$, which correspond to a set of prognostics-driven tasks G^{r}_{progn}. Moreover, each aircraft has a constant average daily utilization rate δ_r.

We formulate the decision-making process for the maintenance of each monitored component as a POMDP. POMDPs have been widely used in the scientific literature for asset management under uncertainty (see [16] and the references therein). We assume that the deterioration process $\{X_t\}_{t\geq 0}$ for every component is a continuous-time discrete-state Markov Chain. It is assumed that each of the components has two unobservable (hidden) working states, $S_X = \{0, 1\}$, where 0 corresponds to the healthy state and 1 corresponds to the degrading state. The observable failure state is defined as state 2, such that the component state space is $S_X \cup 2$. It is noted that a problem with more than 2 working states might be considered. However, this will increase the dimensionality of the problem accordingly. Moreover, from a practical point of view, having two discrete working states makes easier the interpretation of results from the decision-maker and the application of the model in practice [12].

The component is classified as either being in the healthy or degrading state based on a predefined threshold Δ of its true RUL, L_u. The component is classified as healthy and degrading when $L_u > \Delta$ and $L_u \leq \Delta$ respectively. Finally, the component has failed when $L_u = 0$.

The sensor information for every monitored component is passed to the prognostics model and at every decision epoch n and time $t = T_n$, the corresponding RUL predictions, $\overline{L_u(T_n)}$, are obtained. However, the true state of the component cannot be directly inferred from the $\overline{L_u(T_n)}$, because of the uncertainty included in the prediction. To capture the uncertainty of the predictions, the predicted RUL, $\overline{L_u(T_n)}$, is represented by the Gaussian distribution, i.e., $\overline{L_u(T_n)} \sim N(\mu_u, \sigma_u)$. Based on the mean value μ_u and the predefined threshold Δ, we obtain the observed state O_{T_n}. The observed deterioration process $O_{T_n} \in S_O = \{0, 1, 2\}$ is then a discrete-time discrete-state stochastic process. After running the prognostics model at time $t = T_n$, given that the true state of the component $X_{T_n} = k$, $k \in S_X$, if $O_{T_n} = k$, then the state of the system is correctly detected. Otherwise, i.e. if $O_{T_n} \neq k$, the detection is incorrect.

As it is evident, the observed state relates stochastically to the true underlying, but hidden working state of the component, which is either healthy or degrading. This relationship is captured by the state observation matrix $Q = (q_{iz})_{2\times 2}$, where $q_{iz} = P(O_{T_n} = z | X_{T_n} = i)$, $i \in S_X$, $z \in S_O$, is the probability that the decision-maker observes that the component is in state j while the component is in true unobservable state i. We calculate the state observation matrix by using historical data of L_u and $\overline{L_u}$.

Based on the observation O_{T_n} that is received at every decision epoch n, the maintenance planner may choose to perform a maintenance intervention or not, subject to constraints (2)–(7). It is noted that these constraints need to be satisfied for each component individually. Following this, at the first stage of the scheduling model, the set of constraints (1) does not have to be verified, since the computed optimal dates are used as a reference in the second stage of the scheduling framework, when the set of the constraints (1) are considered and specific maintenance dates are allocated to aircraft. The possible maintenance actions are thus defined as $a_{T_n} = \{0 : Do-nothing, 1 : Perform-maintenance\}$.

4.1.2. Update of the state transition law at each decision epoch

The state transition law, $P(X_{T_{n+1}} = j | X_{T_n} = i)$, defines the conditional probability that the component is in the discrete state j at time $t = T_{n+1}$, given that it was at discrete state i at time $t = T_n$, where $i, j \in S_X \cup 2$ and $j \geq i$. Since at every decision epoch new RUL predictions are obtained, we can use these predictions/observations to update the decision-maker's knowledge about the transition rates, and subsequently update his/her knowledge of the degradation process. Recalling that $\Phi(\cdot)$ corresponds to the cdf of normal distribution, we derive the following five expressions:

- if $i = j = 0$:

$$P_{0,0} = P(X_{T_{n+1}} = 0 | X_{T_n} = 0) = \frac{\Phi(\frac{max(\overline{L_u(T_n)})-\mu_u}{\sigma_u}) - \Phi(\frac{\Delta+\delta_r-\mu_u}{\sigma_u})}{\Phi(\frac{max(\overline{L_u(T_n)})-\mu_u}{\sigma_u}) - \Phi(\frac{\Delta-\mu_u}{\sigma_u})} \tag{9}$$

- if $i = j = 1$:

$$P_{1,1} = P(X_{T_{n+1}} = 1 | X_{T_n} = 1) = \frac{\Phi(\frac{\Delta-\mu_u}{\sigma_u}) - \Phi(\frac{\delta_r-\mu_u}{\sigma_u})}{\Phi(\frac{\Delta-\mu_u}{\sigma_u})} \tag{10}$$

- if $i = 0$ and $j = 1$:

$$P_{0,1} = P(X_{T_{n+1}} = 1 | X_{T_n} = 0) = \frac{\Phi(\frac{\Delta+\delta_r-\mu_u}{\sigma_u}) - \Phi(\frac{\Delta-\mu_u}{\sigma_u})}{\Phi(\frac{max\overline{L_u(T_n)}-\mu_u}{\sigma_u}) - \Phi(\frac{\Delta-\mu_u}{\sigma_u})} \quad (11)$$

- if $i = 1$ and $j = 2$:

$$P_{1,2} = P(X_{T_{n+1}} = 2 | X_{T_n} = 1) = \frac{\Phi(\frac{\delta_r-\mu_u}{\sigma_u})}{\Phi(\frac{\Delta-\mu_u}{\sigma_u})} \quad (12)$$

- if $i = 2$ and $j = 2$:

$$P_{2,2} = P(X_{T_{n+1}} = 2 | X_{T_n} = 2) = 1 \quad (13)$$

These expressions formulate the state transition matrix $\mathcal{P} = (p_{ij})_{3\times3}$, $i, j \in S_X \cup 2$.

4.1.3. Belief function

Since the true working state of the component cannot be directly determined by the output of the prognostics model, the state of knowledge of the maintenance planner can be represented by a vector of probabilities called belief state $b_{T_n} = (b^0_{T_n}, b^1_{T_n})$, which is the decision's maker perceived probability of the component being at state 0 and 1 respectively, at time $t = T_n$. According to POMDP theory [24], it can be proven that the whole sequence, or *history*, of observations and actions until time $t = T_n$, $h_{T_n} = \{a_{T_1}, O_{T_1}, \dots .O_{T_{n-1}}, a_{T_n}, O_{T_n}\}$, can be summarized by b_{T_n}. Thus, in order to calculate $b_{T_{n+1}}$, it is sufficient to know b_{T_n}. Therefore, after receiving a new observation $O_{T_{n+1}} = z$ at the next decision epoch $n + 1$, we can calculate, by means of the Bayes rule, the posterior probability vector (or the updated belief), $b_{T_{n+1}}$, whose each element is given by:

$$b^j_{T_{n+1}} = \frac{\sum_{i\in S_X} b^i_{T_n} p_{ij} q_{iz}}{\sum_{i\in S_X}\sum_{l\in S_X} b^i_{T_n} p_{il} q_{lz}}, \qquad j \in S_X \quad (14)$$

4.1.4. Policy and value function

In a fully observable MDP, a policy is a mapping from states to actions. In a POMDP, a policy π is a mapping from belief states to actions. Each policy induces an expected accumulated discounted return. For the current work, the objective of the maintenance planner is to work out a maintenance policy that minimizes the expected long-term maintenance cost of the component over the intended planning horizon $[T_n, T_{end}]$.

The maintenance cost of the component at time $t = T_n$, $T_n \le T_{end}$, is defined as follows:

$$C(T_n) = C^{corr}_u \times (1 - \mathcal{R}(T_n)) + C^{prev}_u \times [\mathcal{R}(T_n) + \frac{\mathbb{E}[\overline{L_u(T_n)}|TSI_u(T_n)] - TSI_u(T_n)}{MTBF_u}] \quad (15)$$

where $TSI_u(T_n) = n \cdot \delta_r$ corresponds to the elapsed time from the installation of the component until $t = T_n$, $MTBF_u$ is the Mean Time Between Failures (estimated from historical data) for the specific component, Finally, $\mathcal{R}(T_n)$ is the component's reliability at time $t = T_n$ and may be interpreted as the probability that the component will not fail until the next decision epoch $n + 1$, i.e., $\mathcal{R}(T_n) = 1 - P^{fail}_u(T_{n+1})$, where the probability of failure at the next decision epoch, $P^{fail}_u(T_{n+1})$, is calculated as follows:

$$P^{fail}_u(T_{n+1}) = \begin{cases} 0 & \text{if} \quad X_{T_n} = 0 \\ P(\overline{L_u(T_n)} < \delta_r) = \Phi(\frac{\delta_r-\mu_u}{\sigma_u}) & \text{if} \quad X_{T_n} = 1 \end{cases} \quad (16)$$

As a result of the chosen action, the maintenance planner receives a total discounted accumulated return:

$$R^u_{T_n} = \sum_{t=T_n}^{T_{end}} \gamma r^u_t \quad (17)$$

where γ is a discount factor and r^u_t is the difference of maintenance cost between two consecutive decision epochs and can be formulated as follows:

$$r^u_t = C(t-1) - C(t) \quad (18)$$

This formulation of the reward function is intended to capture the additional maintenance cost savings or losses that can be incurred because of the decision of the maintenance planner to postpone the maintenance of the component for one additional day. Then, the value function, which can be used to assess the quality of policy π can be written as:

$$V^u_\pi(b_{T_n}) = \mathbb{E}_\pi[R^u_{T_n} | b_{T_n}] \quad (19)$$

and corresponds to the expected return that will be earned over the planning horizon $[T_n, T_{end}]$, starting from belief state b_{T_n}. Among all candidate policies, the one that yields the maximum value function is called the optimal policy, $\pi^\star$:

$$\pi^\star(b_{T_n}) = argmax_\pi V^u_\pi(b_{T_n}) \quad (20)$$

The optimal policy $\pi^\star$ specifies the optimal action to execute at the current decision epoch, assuming that the planner will act optimally in the future. In any POMDP, there is *at least* one optimal policy π^* that achieves the optimal value function [4].

4.2. Prognostics-driven tasks scheduling algorithm

The scheduling algorithm generates a maintenance policy/schedule for each considered component for the intended planning horizon $[T_0, T_{end}]$. It uses a modified version of the POMCP algorithm developed by Silver and Veness [4] to obtain the optimal value function for the maintenance of each component, and thus, define the corresponding optimal maintenance policy for every day of the planning horizon. It should be noted that the term "optimal maintenance policy" that is used in the current study corresponds to the optimal policy defined at each decision epoch based on the received RUL predictions and the update of the knowledge of the state transition rates as calculated in Section 4.1.2. As such, it might not correspond to optimality for the whole maintenance process for the entire life cycle of the monitored component.

For each component, a Monte Carlo Search Tree is built. An example of such tree is visualized in Fig. 3. We assume that the predictions from the prognostics model are available every day, so the decision epoch n corresponds to day n of the planning horizon. Accordingly, the tree is organized in n alternating layers of belief and action nodes, where each layer corresponds to a day of the planning horizon. Each node is characterized by the number of visits N, which counts the number of times this node has been visited, and a value V, which captures the average estimated return of all simulations when starting from this node.

The root node of the tree represents the current updated belief, b_{T_0}, based on the "real" observation O_{T_0}, received by the prognostics model. Instead of updating the belief using Eq. (14), the algorithm uses a particle filter, where each particle corresponds to a sample state. More specifically, to update the belief state, a Monte Carlo procedure is performed, that samples a state from the previous belief state, b_{t-1}, and passes it to a *sample generative model*. The sample generative model, for the given state X_t and for the chosen action $\hat{a}$, provides the successor state X'_t, observation O'_t and reward r_t :

$$\mathcal{G}(X_{t-1}, \hat{a}) = (X'_t, O'_t, r_t) \quad (21)$$

The dynamics of the sample generative model are captured by the POMDP developed in Section 4.1: the successor state X'_t is given by the state transition matrix $\mathcal{P}$ (Section 4.1.2), the sample observation O'_t by the state observation matrix $\mathcal{Q}$ (Section 4.1.1) and the reward r_t by Eqs. (18) & (15). If the sampled observation O'_t matches the real

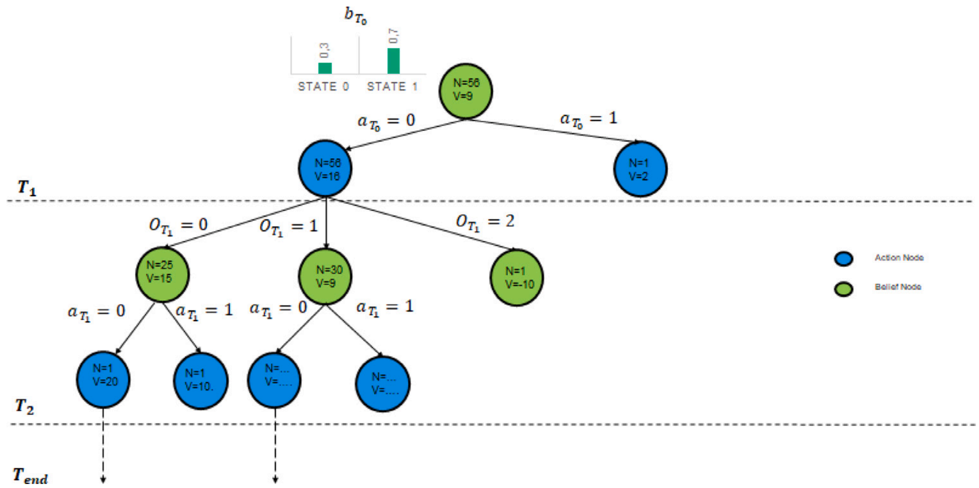

Fig. 3. Monte Carlo component tree.

observation O_t, then state X'_t is added to the set of particles. This procedure is repeated until $\mathcal{K}$ particles have been added to the belief state b_t.

The algorithm starts from the root node of the tree. For $nSim$ episodes a particle is randomly sampled from the belief node b_t and the related sampled state X_t is used as the initial state of the simulation. The available child action nodes depend on whether: (i) there is an available maintenance slot on that specific day of the planning horizon, and (ii) the set of constraints (2)–(7) is verified. If both maintenance actions are available, then an action $\hat{a}$ is selected according to *Upper Confidence Bound for Trees* (UCT) search strategy([25]), defined as follows:

$$\hat{a} = argmax_{a_t}(V(b_t a_t) + c\sqrt{\frac{\log(N(b_t))}{N(b_t a_t)}}) \tag{22}$$

where the constant c determines the exploration-to-exploitation ratio. This action is then passed to the sample generative model that determines the successor state X'_{t+1}, observation O'_{t+1}, and reward r_{t+1}. At the succeeding belief nodes, the observation O_{t+1} determines which branch of the tree the algorithm needs to follow. The same procedure repeats until the desired planning horizon or a terminal state is reached (the component has failed or the component is scheduled for maintenance), or an unexplored belief node is encountered.

In the latter case, the tree is expanded by precisely one node, corresponding to the new belief state encountered during that simulation. Then a random simulation is run, where actions are selected according to a rollout policy, like uniform random action selection. This process is repeated until the terminal state or the end of the desired planning horizon has been reached. The rewards obtained through this simulation step are backpropagated through the internal nodes upwards in the tree, defining for each action of the root node an approximated expected discounted reward. When all the simulations are complete, the algorithm selects the action node with the greatest value of the value function. Following the formulation of the value function in accordance with Eqs. (15)–(19), the algorithm chooses the action that minimizes the expected long-run maintenance cost. When a real prediction/observation from the prognostics has been received, we prune the tree at the belief node determined by the received observation. This specific belief node becomes the new root node of the tree and, as such, all the other belief nodes are now impossible.

Moreover, it is reminded that our goal is to derive a maintenance policy for the whole intended planning horizon. However, up to this point, the algorithm has only chosen the optimal action at the root node, which corresponds to the present day. To plan for the rest of the days of the planning horizon, without having to wait for a new prediction/real observation from the prognostics, we prune the tree at the most probable observation expected to be received on the next day of the planning horizon, i.e., at the succeeding belief node with the highest amount of visits N. For example, in Fig. 3, in order to plan for T_1, we prune the tree at $O_{T_1} = 1$ as it has more visits than $O_{T_1} = 0$ and $O_{T_1} = 2$. This belief node becomes then the new root node of the tree and the algorithm is run again to generate the optimal action for this specific day of the planning horizon. The same process repeats until we reach the end of the planning horizon. The pseudocode for the prognostics-driven tasks scheduling algorithm can be found in Appendix A. Readers are kindly referred to [4] for a more detailed explanation of the POMCP algorithm.

The final output of the algorithm is a sequence of the optimal maintenance actions corresponding to each day of the planning horizon, which constitutes the maintenance schedule for the specific component. The day the component is scheduled for maintenance corresponds to the optimal maintenance date for this specific component individually and defines the due date of the corresponding prognostics-driven task. The defined prognostics-driven tasks will be passed to the Deep Reinforcement Learning algorithm, in order to be considered for scheduling, together with the preventive and corrective maintenance tasks, at the aircraft level.

5. Aircraft fleet maintenance scheduling

The aircraft fleet maintenance scheduling algorithm presented in this paper outputs a maintenance schedule which details the allocation of the tasks and the corresponding aircraft to the available maintenance slots over the desired planning horizon. We formulate the aircraft fleet maintenance scheduling problem as a sequential decision-making process that can be modeled through a Markov Decision Process (MDP). An MDP is defined on the basis of a state space, action space, and reward function. However, constructing a state and action space by considering all the aircraft in the fleet and solving the resulting MDP, even with a DRL approach, can require too much computational time to be suitable for operational use.

Inspired by real airline practice, we reduce the problem size by constructing an MDP only for the tasks and the corresponding aircraft that is not possible to be allocated to the fixed maintenance slots. In order to know these tasks, we have to solve the task allocation problem for the fixed slots. Solving this type of problem is fairly simple, as in fixed maintenance slots the aircraft registrations are pre-defined and the aircraft will be grounded for maintenance anyway as part of a major letter check, meaning that the maintenance planner does not have to worry either about matching the duration of the task with the duration of the slot or about causing potential schedule changes, as in the case of flexible maintenance slots. We use a *Greedy algorithm* to

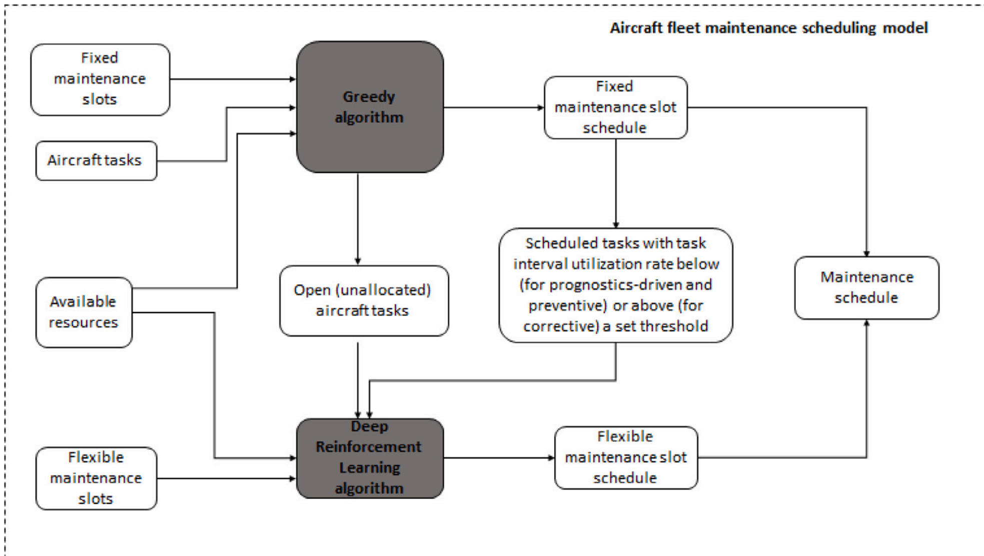

Fig. 4. Aircraft fleet scheduling model overview.

assign the considered aircraft maintenance tasks to the available fixed maintenance slots, subject to the set of constraints (2)–(7). The tasks that could not be allocated to the fixed maintenance slots, together with (i) the preventive and prognostics-driven tasks and (ii) the corrective tasks, that have been scheduled at the fixed slots but have a task interval utilization rate below and above a set threshold, th_{prev} and th_{corr} respectively, are then passed to the *Deep Reinforcement Learning* algorithm, which assigns these tasks to the flexible maintenance slots, satisfying the four objectives described in Section 3.4 and subject to the set of constraints (1)–(7). The fixed and flexible slots' maintenance schedule are then combined to generate the final maintenance schedule for the aircraft fleet.

An overview of the aircraft fleet maintenance scheduling algorithm is presented in Fig. 4. In the next sections, a detailed description of the developed algorithms is provided.

5.1. Greedy algorithm

The objective of the greedy algorithm is to allocate as many tasks as possible to the fixed maintenance slots, while achieving an efficient task interval utilization. First, the maintenance tasks of each aircraft are sorted in ascending order according to the number of maintenance opportunities/fixed slots left for the execution of each task, that verify the set of constraints (2)–(7). The tasks that have no remaining maintenance opportunities, i.e., it is not possible to be assigned to the fixed slots, are removed from the list and added to the pool of tasks that will be considered for scheduling by the DRL algorithm.

The algorithm then selects the first element from the list, that corresponds to the most urgent task. If the task is either prognostics-driven or preventive, it is assigned to the latest possible fixed slot. In case it corresponds to a corrective task, it is assigned in the earliest possible fixed slot. After the task is assigned, the available workhours per skill in the used maintenance slot are reduced by the amount of workhours per skill required for the execution of the task. The tasks are then sorted again in ascending order according to the number of the updated remaining fixed slots, and the same process repeats until all aircraft tasks have been considered.

The final output of the algorithm consists of a maintenance schedule, detailing the allocation of tasks to the fixed maintenance slots, and a list of tasks to be considered for scheduling in the flexible slots by the DRL algorithm. The pseudocode for the greedy algorithm can be found in Appendix B.

5.2. Deep Reinforcement Learning (DRL) algorithm

In the Reinforcement Learning context, an agent learns how to interact with the surrounding environment by following a specific policy. In the basic representation, at every decision point t, an agent observes a state $s_t \in \mathcal{S}$, chooses an action $a_t \in \mathcal{A}$ according to the policy $\pi(\mathcal{S} \to \mathcal{A})$, and receives an immediate reward $r_t \in \mathcal{R}$. The objective of the agent is to determine the optimal policy $\pi^\star$ that maximizes the expected sum of long-term discounted rewards. In the next sections, a detailed formulation of the state and action space, the reward function, as well as the DRL algorithm is provided.

5.2.1. State space

The decision-making process occurs in a sequential manner, i.e., at each flexible maintenance slot m, $m \in M_{Flexible}$, the DRL agent has to decide which aircraft to schedule. The state of the environment in every maintenance slot that influences this decision, is captured by the state vector:

$$s_m = \langle s_r, ND_m, DT_m, SC_m, STN_m \rangle \tag{23}$$

where s_r is defined as follows:

$$\begin{aligned} s_r = \{ & OT_r, TT_r, SU_r, FS_r, RU_r, NRU_r, DR_r, FNS_r, \\ & MUR_r, MDR_r, RFS_r, RO_r, TN_r \} \end{aligned} \tag{24}$$

Each feature of the state vector s_r contains indicators about the considered tasks of aircraft r, $r \in R$, with respect to the current considered flexible maintenance slot m. Also, these features are designed, through trial-and-error experimentation, in a way such that the planning objectives described in Section 3.4 are captured, as part of the environment. Before presenting the details of the features, some notations should be given in advance. We define the task interval utilization rate, UR_g, as the moment in time a task is executed relative to the length of its interval. The length of the task interval refers to the due date of the task, Due_g, in the case of corrective and preventive maintenance tasks, or the optimal maintenance date defined by the POMCP scheduling algorithm, OD_g, in the case of prognostics-driven tasks. Then, the task interval utilization rate can be calculated as follows:

$$UR_g = \begin{cases} \frac{Start_m - Arrival_g}{Due_g - Arrival_g}, & \text{if} \quad g \in G_{corr} \cup G_{prev} \\ \frac{Start_m - Arrival_g}{OD_g - Arrival_g}, & \text{if} \quad g \in G_{progn} \end{cases} \tag{25}$$

Based on the notations above, the state features are defined as follows:

- $\mathbf{OT_r}$: The open tasks of aircraft r when considering slot m. If there are no open tasks, the corresponding aircraft features are removed from the state vector.
- $\mathbf{TT_r}$: A binary variable that indicates the type of the open tasks of aircraft r. Unitary if only corrective maintenance tasks are considered, zero if either preventive or prognostics-driven tasks are considered as well.
- $\mathbf{SU_r}$: A binary variable that indicates whether aircraft r can be assigned in slot m, based on the verification of the set of constraints (2)–(7). Unitary if aircraft r can be assigned in slot m.
- $\mathbf{FS_r}$: A binary variable that indicates whether all open tasks of aircraft r can be assigned in the current maintenance slot m. Unitary if all open tasks can be assigned to this maintenance slot, verifying the set of constraints (2)–(7).
- $\mathbf{RU_r}$: The achieved average utilization rate of the prognostics-driven and preventive maintenance tasks, if aircraft r is assigned in the maintenance slot m. Based on Eq. (32), the average utilization rate for aircraft r can be defined as follows:

$$RU_r = \frac{\sum_{g=1}^{K} UR_g}{K}, \quad g \in G^r_{prev} \cup G^r_{progn}, \quad K = |G^r_{prev}| + |G^r_{progn}| \tag{26}$$

- $\mathbf{NRU_r}$: The achieved average utilization rate of the corrective maintenance tasks, if aircraft r is assigned in the maintenance slot m. Based on Eq. (32), the average utilization rate for aircraft r can be defined as follows:

$$NRU_r = \frac{\sum_{g=1}^{K} UR_g}{K}, \quad g \in G^r_{corr}, \quad K = |G^r_{corr}| \tag{27}$$

- $\mathbf{DR_r}$: The ratio of the task with the highest duration to the duration of the maintenance slot:

$$DR_r = \frac{max(Duration_g)}{Duration_m}, \quad g \in G^r. \tag{28}$$

- $\mathbf{FNS_r}$: A binary variable indicating whether there is any flexible maintenance slot in the future where all the open tasks can be addressed, subject to the set of constraints (2)–(7). Unitary if such a slot exists.
- $\mathbf{MU_r}$: The maximum average utilization rate, RU_r, that can be achieved for the preventive and prognostics-driven maintenance tasks, if they are assigned to the forthcoming available flexible maintenance slots, subject to the set of constraints (2)–(7).
- $\mathbf{MDR_r}$: The maximum DR_r that can be achieved if the open tasks of aircraft r are assigned to the forthcoming available flexible maintenance slots, subject to the set of constraints (2)–(7)
- $\mathbf{RFS_r}$: The total number of flexible maintenance slots, where all the open tasks of aircraft r can be addressed, subject to the set of constraints (2)–(7).
- $\mathbf{RO_r}$: The number of remaining maintenance opportunities/ available flexible maintenance slots for aircraft r, before the first task goes due, subject to the set of constraints (2)–(7).
- $\mathbf{TN_r}$: The aircraft registration number.
- $\mathbf{ND_m}$: The difference, captured in days, between the start date of the slot and the start date of the planning horizon.
- $\mathbf{DT_m}$: The total number of due tasks across the aircraft fleet until flexible maintenance slot m.
- $\mathbf{SC_m}$: The total number of changes performed in the maintenance schedule until flexible maintenance slot m.
- **STN**: The aircraft registration number that was assigned for maintenance in flexible maintenance slot m in the existing maintenance schedule.

5.2.2. Action space

At each maintenance opportunity m, the maintenance planner needs to decide which aircraft to schedule for maintenance. The action space at maintenance slot m can be defined as follows:

$$\mathcal{A}_m(s_m) = (r \in R, STN, NULL) \tag{29}$$

Action $r \in R$ refers to allocating the corresponding aircraft for maintenance in maintenance slot m. The STN action corresponds to selecting for maintenance the aircraft that was originally allocated in maintenance slot m in the existing schedule, meaning that no schedule change is observed if this action is chosen. The $NULL$ action corresponds to not scheduling any of the considered aircraft in maintenance slot m.

When the agent selects an aircraft for maintenance, it sorts the aircraft open maintenance tasks in ascending order, according to the remaining maintenance opportunities of each task. Then it starts allocating tasks to the maintenance slot by considering them in the order they are referenced in the sorted list, as long as the set of constraints (1)–(7) is verified. After all the open maintenance tasks are considered for scheduling, the agent then tries to reschedule the preventive and prognostics-driven tasks, and the corrective tasks, that have been allocated to fixed maintenance slots, but have a task interval utilization rate below and above a set threshold respectively. These tasks are sorted in ascending order, according to the task interval utilization rate. Afterward, the agent starts assigning these tasks to the maintenance slot by considering them in the order they are referenced in the sorted list, as long as the set of constraints (1)–(7) is verified.

The $NULL$ action corresponds to not scheduling any of the considered aircraft on maintenance slot m.

5.2.3. Reward function

The reward function is of paramount importance, as it quantifies the performance of each chosen action with respect to the environment. Since each action should be chosen according to the objectives of the maintenance planner as defined in Section 3.4, the reward function is formulated as a weighted combination of critical state features at state s_m, including the number of due the tasks, DT_m, the ratio of ground time, DR_r, the schedule changes, SC_m, and task utilization, RU_r and NRU_r. The procedure for calculating the reward function r_m is explained in details in Appendix C.

5.2.4. Deep Q learning network

A widely used Reinforcement Learning algorithm is Deep Q Network (DQN). This concept uses a multi-layered neural network with weights θ to approximate the Q value function. The inputs to the neural network are the state features, while the output consists of the Q function value for each state–action pair. By making use of the neural network, the algorithm can handle more complicated decision problems with large state–action spaces.

The DQN used in this paper is a deep neural network consisting of five fully connected layers with one input layer, three hidden layers and one output layer. The structure of the DQN is illustrated in Fig. 5. In order to prevent having a varying input and output layer size due to the different number of aircraft with open tasks at every decision point, we choose to always consider at every decision point the 10 most critical aircraft in terms of RO_r. That is, we sort all the aircraft in ascending order according to the remaining opportunities (available flexible maintenance slots) until their first task goes due and we select the first 10 aircraft from this sorted list. The reason lies in the fact that is very unlike that the optimal aircraft to be allocated in the maintenance slot m is not within this 10 most critical aircraft list. This results in a number of 134 nodes for the input layer and 12 nodes for the output layer. Also, each hidden layer consists of 64 neurons. We use the ReLu activation function for the input and the hidden layers and the linear activation function for the output layer.

6. Computational experiments

6.1. Case study

To verify and demonstrate the proposed two-stage scheduling framework, we perform a case study for a fleet of 34 wide-body aircraft of

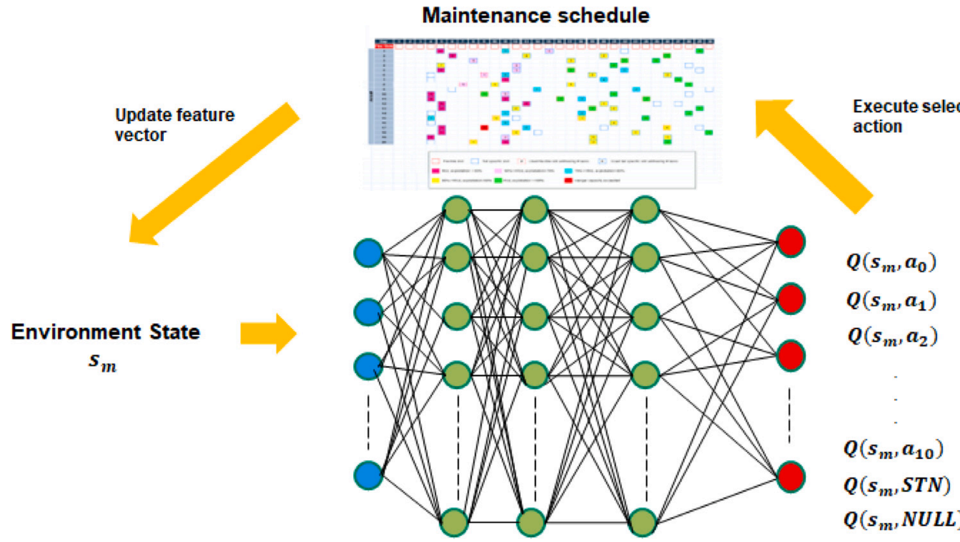

Fig. 5. DQN structure.

different types from a major European airline. Each aircraft has a set of open preventive and corrective maintenance tasks according to the MPD and MEL, which are updated on a daily basis. In total, we consider 1517 open corrective and preventive maintenance tasks, spreading over a period of 4 months of the airline operations.

Moreover, we assume that 10 aircraft from the fleet are equipped with sensors that monitor 25 non-safety critical components from each aircraft on a permanent basis. For every monitored component, we simulate RUL predictions which are updated on a daily basis and are assumed to follow the normal distribution. The working state of the components included in the 10 aircraft is different, i.e., every component has a different true RUL.

All the former tasks can be addressed in dedicated maintenance opportunities, for which the availability and the resources change over time. The workforce is divided into multiple skills with specific availability and is organized into three shifts per day. Data about the corrective and preventive maintenance tasks, the maintenance slots schedule, and the available resources were provided by the airline.

The analysis of the case study is divided into five parts: First, the simulation setup and approach are described. Secondly, the results of the prognostics-driven tasks scheduling algorithm are presented, in order to evaluate the efficiency of the 1st stage of the framework with respect to scheduling the maintenance of the monitored components. This is followed by a sensitivity analysis of the preventive-to-corrective maintenance cost ratio, to analyze its impact on the number of tasks going due and the RUL exploitation of the monitored components. Third, a stand-alone performance evaluation of the 2nd stage of the scheduling framework, i.e. the DRL algorithm, including only the preventive and the corrective maintenance tasks, is performed against the executed maintenance schedule of our partner airline. In the fourth part, the results obtained after running the full version of the two-stage scheduling framework, i.e., including the prognostics-driven, the preventive, and the corrective tasks, are presented. The last part of the case study focuses on a cost–benefit analysis of a Corrective vs a CBM approach.

6.1.1. Simulation of RUL predictions

We simulate the RUL predictions by applying the Support Vector Regression (SVR) prognostics model developed in [26] on the C-MAPSS dataset [27] for turbo-fan engines. We further assume that the time cycles used in the C-MAPPS dataset correspond to Flight Cycles (FCs). Following the approach described in [28], we organize the obtained predictions in 4 clusters based on the prediction accuracy and uncertainty, described by MAE and standard deviation σ in FCs respectively:

Table 4
State observation matrix for components belonging in Cluster # 1.

X \ O	**0** (Healthy)	**1** (Degrading)
0 (Healthy)	0.99	0.01
1 (Degrading)	0.14	0.86

Table 5
State observation matrix for components belonging in Cluster # 2.

X \ O	**0** (Healthy)	**1** (Degrading)
0 (Healthy)	1	0
1 (Degrading)	0.22	0.78

Table 6
State observation matrix for components belonging in Cluster # 3.

X \ O	**0** (Healthy)	**1** (Degrading)
0 (Healthy)	0.99	0.01
1 (Degrading)	0.31	0.69

Table 7
State observation matrix for components belonging in Cluster # 4.

X \ O	**0** (Healthy)	**1** (Degrading)
0 (Healthy)	0.96	0.04
1 (Degrading)	0.39	0.61

- **Cluster #1**: $MAE \sim 8.32$ and $\sigma \sim 12.24$
- **Cluster #2**: $MAE \sim 13.89$ and $\sigma \sim 19.69$
- **Cluster #3**: $MAE \sim 23.52$ and $\sigma \sim 27.35$
- **Cluster #4**: $MAE \sim 37.28$ and $\sigma \sim 40.93$

We then assume that for the 25% of the components (7 components in total) the computed daily RUL predictions belong to Cluster 1, for 25% to Cluster 2 (6 components in total), for 25% to Cluster 3 (6 components in total) and for 25% to Cluster 4 (6 components in total). Furthermore, we set the threshold $\Delta = 40$ FCs to differentiate between the healthy and the degrading state. We chose this value for Δ on the basis that, according to a daily flight utilization of $\delta_r = 4$ FCs, it provides the maintenance planner with at least 10 days to schedule the

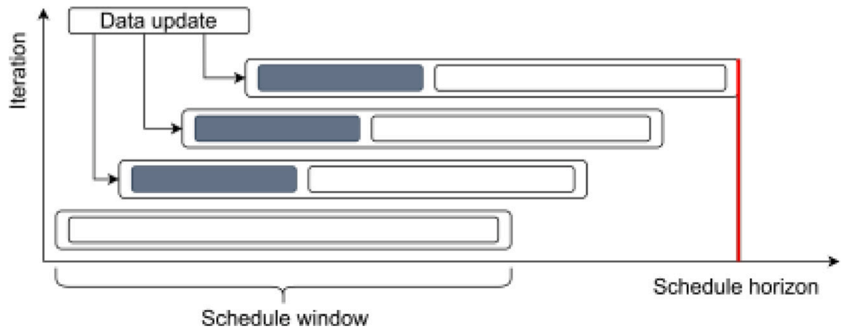

Fig. 6. Rolling horizon approach [9].

maintenance of the component, without inducing unnecessary changes to the maintenance schedule. The calculated state observation matrices that describe the stochastic relationship between the true, $X \in S_X$, and the observed, $O \in S_O$, working state of the component for each cluster are as follows (see Tables 4–7).

6.1.2. Simulation of new task information arrival

In order to simulate the dynamic process of the arrival of new corrective and preventive tasks, and also the update of RUL predictions for the monitored components, we implement a rolling horizon approach (see Fig. 6). A maintenance schedule is generated for a fixed time window. Afterwards, the planning horizon shifts one day ahead, where new tasks may arrive and/or new RUL predictions are obtained. However, the new information about the maintenance tasks may compromise the feasibility of the existing schedule. Our framework takes rescheduling actions to produce a new and feasible schedule for the intended time window, while also minimizing the number of schedule changes in the next 3 days (highlighted gray area in Fig. 6), i.e., $CP = 3$. This value was chosen on the basis that in our partner airline, aircraft registrations are assigned to flights 3 days upfront. The same process repeats until the end of the planning horizon is reached. It is noted that no scheduling opportunities beyond the end of the planning horizon are considered.

6.1.3. Assumptions

- **Aircraft utilization is known and constant**. The daily aircraft utilization δ_r, $r \in R$ is set to 15 FHs - 4 FCs, according to historical aircraft utilization values of an airline.
- **There are no component-wise dependencies**. The potential economic, stochastic and structural dependencies of the components are not considered.
- **RUL predictions are obtained on a daily basis**. The SVR prognostics model produces every day one RUL prediction for every monitored component.
- **RUL predictions follow a normal distribution**.
- **The monitored components are not critical for the safe and reliable operation of the aircraft.**
- **The impact of the environment on the deterioration of the components is not considered.**

6.2. Results analysis of the prognostics-driven tasks maintenance scheduling model

In this section, we evaluate the performance of the first stage of the framework, i.e., we assess the scheduling of the 250 prognostics-driven tasks individually, not considering the preventive and corrective maintenance tasks of the aircraft fleet. The parameters of the POMCP algorithm are defined through experimentation in the following Table 8:

Table 8
Parameters of the POMCP algorithm.

Parameter	Value
exploration constant c	100
# belief particles	1200
# episodes	500

Moreover, the corrective and preventive replacement cost of every monitored component is set to $C_u^{Corr} = 25{,}000$ and $C_u^{Prev} = 10{,}000$ respectively. The magnitude of the cost values was driven by Freeman et al. [29], where average historical values of true preventive and corrective repair costs were used. Based on the daily updated RUL predictions simulated according to Section 6.1.1, the prognostics-driven tasks scheduling model generates at each day the maintenance schedule of each monitored component for the next 30 days of the planning horizon. It is noted that at this stage we are only interested in scheduling efficiently the prognostics-driven tasks, which means that up to this point no decisions at aircraft level are taken. As such, we are only interested in executing maintenance in the monitored component on-time, while at the same time, achieving the highest possible RUL exploitation, in a computational efficient manner. Following this, we do not consider the effect of the chosen maintenance action with respect to the aircraft operational availability and potential schedule changes. The proposed scheduling model is implemented in Python 3.7.6, on a laptop computer with Intel Core i5 processor 9300, 16 GB RAM, and NVIDIA 1660Ti GPU. The results for different values of the discount factor γ are summarized in Table 9.

Table 9
Results of the prognostics-driven tasks scheduling model for different values of discount factor γ.

Discount factor γ	Due tasks	Average RUL exploitation of the components	Computational time (s) per task
0.3	40	90.1%	3.4
0.5	33	89.6%	5.1
0.7	30	84.2%	15.7
0.9	28	79.4%	31.3

It can be observed that increasing the discount factor γ leads to a decrease in the number of prognostics-driven tasks going due, while at the same time, there is also a decreasing trend in the RUL exploitation of the components. This happens because a higher discount factor places a larger emphasis on the future long-term rewards, and as such a more conservative scheduling approach is followed. The computational time also increases with the increase of the discount factor, as the POMCP algorithm is encouraged to explore actions and observations that are further into the future, leading to longer branches within the POMCP search tree. Since in this research, we are interested in achieving good results both in terms of timely task scheduling and high RUL exploitation, but also in a computationally efficient manner, we decided to adopt a value of 0.5 for the discount factor for use by the two-stage scheduling framework evaluated in the next section. However, if computational efficiency is not a requirement or more emphasis is placed on preventing tasks from going due than exploiting the RUL of the components, then adopting a higher value is recommended.

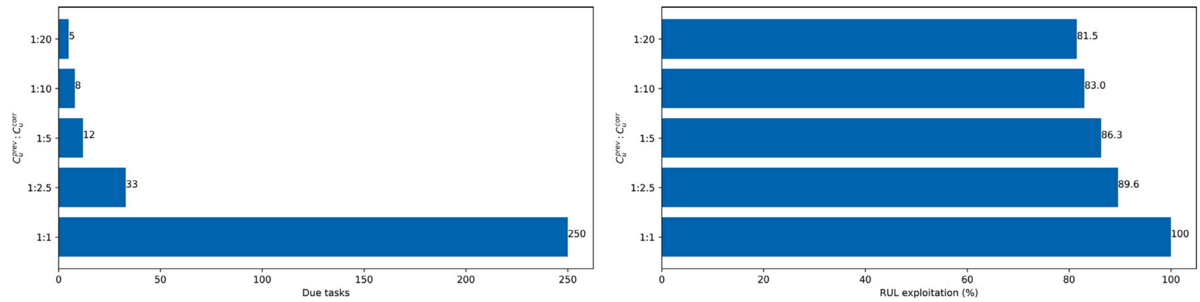

Fig. 7. Due prognostics-driven tasks and RUL exploitation for different ratios of $C_u^{prev} : C_u^{corr}$.

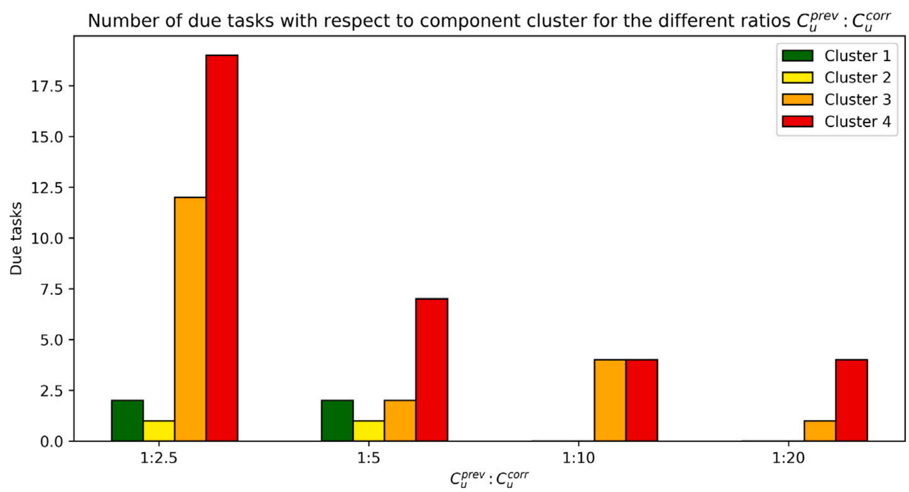

Fig. 8. Due tasks per cluster of components.

6.2.1. Sensitivity analysis of the preventive to corrective maintenance cost ratio

We also performed a sensitivity analysis of the scheduling performance of the prognostics-driven tasks scheduling model by changing the ratio of the preventive to corrective maintenance cost, $C_u^{prev} : C_u^{corr}$. As noted before, we have assumed a discount factor $\gamma = 0.5$. Hence, we run the first stage of the proposed framework for different scenarios of preventive and corrective maintenance cost ratios. In Fig. 7, the number of the due tasks and RUL exploitation are presented for a range of preventive to corrective maintenance cost ratios.

As expected, in the extreme scenario when the $C_u^{prev} : C_u^{corr} = 1 : 1$, the model does not schedule any component for maintenance, since there is no cost benefit for preventively maintaining the component. Across the other tested scenarios, it can be seen that increasing the preventive to corrective cost ratio causes a decrease in the number of the due prognostic-driven tasks, while at the same time, average RUL exploitation decreases. This observed trade-off between timely component maintenance execution and RUL exploitation is expected, since a higher corrective maintenance cost relates to a more conservative scheduling approach. In other words, aiming for a higher RUL exploitation becomes less attractive, as the risk of exceeding the RUL and repairing the component induces very high corrective maintenance costs.

Finally, in Fig. 8, we present an analysis of the number of components from each cluster that were not scheduled on time in every scenario. Once more, it can be observed that the majority of the components across all scenarios that were not scheduled on-time, come from Clusters #3 and #4.

6.3. Results analysis of the aircraft fleet maintenance scheduling model

In this section, we evaluate the performance of the aircraft fleet scheduling model. As explained in Section 5.2, the DRL algorithm requires a set of input parameters. These input parameters, which were defined through trial-and-error experimentation, are presented in Table 10:

Table 10
Parameters of the DQN.

Parameter	Value
W_{fail}	-10^6
W_{resch}	-2×10^5
W_{ground}	10^5
W_{util}	5×10^4
Initial exploration rate (ϵ_0)	1.0
Exploration rate decay ($\frac{d\epsilon}{dt}$)	0.995
Final exploration rate (ϵ_T)	0.001
Batch size	64
Hidden Layers	3
Dense size (neurons)	64
Training episodes	10000

The proposed DQN is implemented on the same laptop described above. It is trained on 10 days of historical maintenance data provided by the airline and it makes use of a schedule window of 120 days. The training curve for 10,000 episodes is visualized in Fig. 9. It can be seen that the sum of rewards increases with the increase of training steps, which indicates that the proposed DQN has learned the proper maintenance decisions for different situations in an efficient way.

In order to benchmark the performance of the DRL scheduling algorithm, a comparison between the actual airline maintenance schedule for the considered planning period and the schedule produced by the DRL algorithm is performed. However, for a fair comparison, the prognostics-driven tasks are excluded, since the airline has not yet included such type of tasks in its maintenance practice. The model would be evaluated on the basis of the four planning objectives described in

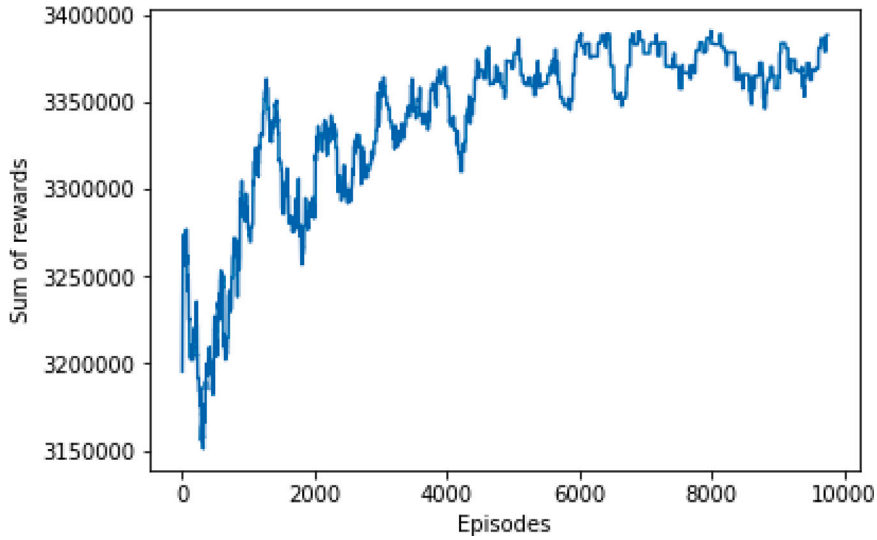

Fig. 9. Sum of rewards during agent training.

Table 11
Number of due tasks and ground time for the airline and the DRL algorithm.

Schedule method	Due tasks	Scheduled tasks	Used slots	Total ground time (hrs)
Airline Schedule	0	1517	248	4,345.43
DRL algorithm	0	1517	243	4,170.39

Table 12
Number of schedule changes and task interval utilization for the airline and the DRL algorithm.

Schedule method	Schedule changes	Preventive tasks utilization	Corrective tasks utilization
Airline Schedule	43	74.86%	53.26%
DRL algorithm	22	75.11%	52.03%

Section 3.4. The thresholds, th_{prev} and th_{corr}, that define the preventive and corrective tasks to be considered for rescheduling by the DRL algorithm, are set to 0.8 and 0.5 respectively.

Table 11 summarizes the results in terms of due tasks and total ground time used, which correspond to the planning objectives of task execution and aircraft operational availability. In both cases, no tasks went due. Additionally, in the maintenance schedule produced by the DRL algorithm, 5 less maintenance slots are used compared to the airline schedule. Moreover, the DRL algorithm saves 175.04 h or 7.29 days of ground time with respect to the actual airline schedule. Whereas the airline schedule makes use of slots with an average duration of 17.5 h, the DRL algorithm makes use of slots with an average duration of 17.1 h. All the former indicate that the DRL algorithm is able to schedule tasks more effectively than the airline.

In Table 12 the results with respect the two remaining planning objectives, i.e., schedule stability and task interval utilization, are presented. DRL algorithm requires significantly fewer last-notice changes, which indicates that the model can produce more stable schedules in comparison with the airline. Moreover, the DRL model achieves better task interval utilization for both the corrective and the preventive maintenance tasks. In other words, the model is able to schedule corrective tasks as further away as possible from their due date and preventive tasks as close as possible to their due date.

Finally, the average computational time needed for every day of the planning horizon was ~ 5.9 s, which makes the DRL scheduling model suitable for ad-hoc decision-making in a real airline maintenance environment.

6.4. Results analysis of the two-stage scheduling framework

In this section, we run the full version of the two-stage scheduling framework, i.e., we consider the 250 prognostics-driven tasks, together with the 1517 corrective and preventive maintenance tasks. We use the results of the prognostics-driven tasks scheduling model from Section 6.2, to simulate the updated recommended maintenance dates for the monitored components for every day of the planning horizon.

The results are summarized in Table 13. The proposed framework schedules all corrective and preventive maintenance tasks on-time, while it also manages to schedule 241 out of 250 prognostics-driven tasks. It is observed that the framework manages to schedule on-time more prognostics-driven tasks than the prognostics-driven tasks scheduling algorithm by itself (241 vs 217-see Section 6.2). This is due to the fact that the prognostics-driven tasks are now considered together with the preventive and corrective maintenance tasks. The algorithm tries to bundle all these types of tasks together and schedule them in the minimum possible amount of maintenance slots, while also minimizing the schedule changes. As a result, it follows a more conservative approach with respect to the allocation of the prognostics-driven tasks, which is reflected in the achieved prognostics-driven task utilization (70.5%). This value is lower than the one achieved by the prognostics-tasks scheduling algorithm when run in isolation from the framework (89.6% - see Section 6.2), verifying the more conservative scheduling of the prognostics-driven tasks in exchange for fewer tasks going due.

Moreover, we perform an analysis of the components that correspond to the prognostics-driven tasks that went due with respect to their cluster category (Fig. 10). We observe that the majority of the components belong to cluster #4, which is characterized by RUL predictions with the higher MAE and standard deviation.

Also, an increase in the number of used slots and total ground time is observed in comparison to the results achieved when no prognostics-driven tasks were considered, but this is due to the additional ground time requirements of the introduced prognostics-driven tasks. Finally,

Table 13
Results of the two-stage scheduling framework.

Planning objectives	Performance indicators	Model	Airline[a]
Task execution	Due tasks (prognostics-driven)	9	N/A
	Due tasks (corrective and preventive)	0	0
	Scheduled tasks	1758	1517
Aircraft operational availability	Used slots	274	248
	Total ground time (hrs)	4,449.3	4345.4
Schedule stability	Schedule Changes	44	43
Task interval utilization	Corrective task utilization	49.53%	53.26%
	Preventive task utilization	75.62%	74.86%
	Prognostics-driven task utilization	70.51%	N/A

[a]Airline results without the 250 prognostics-driven tasks.

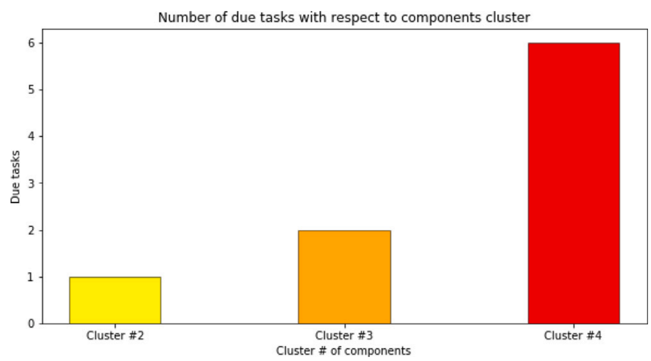

Fig. 10. Due tasks per cluster of components.

due to the introduced uncertainty in the RUL predictions, an increase in the number of schedule changes is observed. The schedule changes caused by the prognostics-driven tasks account for 54% of the total schedule changes. An analysis of the number of the components from each cluster that contributed to the schedule changes is visualized in Fig. 11. The majority (58%) of the prognostics-driven tasks related schedule changes are once more due to components belonging to cluster #4, which is characterized by RUL predictions with the highest degree of uncertainty. In practice, this means that an airline, having components with historical values of RUL predictions falling within these levels of accuracy and uncertainty, should consider incorporating additional slot flexibility.

The average computational time of running the two-stage scheduling framework, i.e., the POMCP and the DRL algorithm, for every day of the planning horizon is approximately ~ 10.9 sec, which highlights the suitability of the framework for application in a real airline environment. This computation time was obtained using the same laptop described before.

Finally, readers are referred to [30] for a detailed comparison of the efficacy of the DRL algorithm against a MILP algorithm, that is used instead in the second stage of the scheduling framework. The performance of both algorithms is evaluated against three real maintenance scenarios in a CBM context for different aircraft fleet sizes with data provided by our partner airline, but also enriched (in a similar fashion with this study) with simulated data for prognostics-driven tasks from the C-MAPSS dataset. The study results highlight that the DRL approach can produce both efficient and stable maintenance schedules. At the same time, it demonstrates an increased computational efficiency which stays relatively unaffected by the problem size and the number of considered variables.

6.4.1. Cost–benefit analysis of CBM vs corrective maintenance

In this section, a cost–benefit analysis of the CBM concept is performed, assuming that the 250 introduced prognostics-driven tasks replace tasks that were addressed through the Corrective Maintenance approach. According to the Corrective Maintenance approach, the component is replaced only after it fails and thus the total maintenance cost, C_{CM}, is calculated as follows:

$$C_{CM} = \sum_{r \in R} \sum_{u \in U} C_u^{corr} \tag{30}$$

where $C_u^{corr} = 25{,}000$, as defined in Section 6.2. According to the CBM approach, the maintenance cost, C_{CBM}, is formulated as follows:

$$C_{CBM} = \sum_{r \in R} \sum_{u \in U} C_u^{CBM} \tag{31}$$

where:

$$C_u^{CBM} = \begin{cases} C_u^{corr}, & \text{if} \quad SD_u > EoL_u \\ C_u^{prev} \cdot \left(1 + \frac{EoL_u - SD_u}{EoL_u}\right) & \text{if} \quad SD_u \leq EoL_u \end{cases} \tag{32}$$

SD_u is the scheduled maintenance date and the for component u, EoL_u is the date that corresponds to the end of life of the component and $C_u^{prev} = 10{,}000$, as defined in Section 6. The comparison of the maintenance costs induced when following the Corrective Maintenance approach and when implementing the CBM approach as defined by the two-stage scheduling framework is visualized in Fig. 12. The results demonstrate that the CBM two-stage scheduling framework is beneficial, as it can lead to a reduction of maintenance costs as high as 46.2% compared to a Corrective Maintenance approach. However, this cost–benefit analysis does not include the costs of sensor installation and certification, or the costs related to the development of the prognostics-driven models.

7. Conclusions

In this paper, we presented a novel two-stage CBM scheduling framework for a fleet of aircraft in a disruptive environment. The RUL prognostics, are updated on a daily basis with new sensor measurements and are characterized by uncertainty which follows the normal distribution. On top of that, also the list of preventive and corrective maintenance tasks is continuously updated. The maintenance planning model takes into account the list of different types of maintenance tasks, along with available maintenance slots, the available resources, and the existing maintenance schedule, to produce the maintenance schedule of the aircraft fleet using a rolling horizon approach. The overarching goal is to prevent tasks from going due, while at the same time, ensuring high fleet availability, schedule stability, and efficient task interval utilization.

The proposed methodology follows a POMDP approach, incorporating two scheduling blocks. The first scheduling block is based on a modified version of the POMCP algorithm to derive the optimal maintenance policy at component level. The second scheduling block uses a DRL approach to produce a maintenance schedule at the aircraft fleet level.

The performance of the proposed model was evaluated on a real case study from our partner airline for a fleet of 34 wide-body aircraft

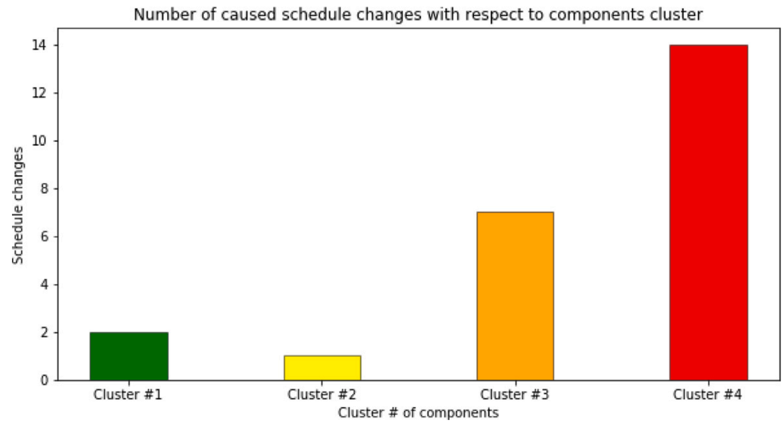

Fig. 11. Schedule changes per cluster of components.

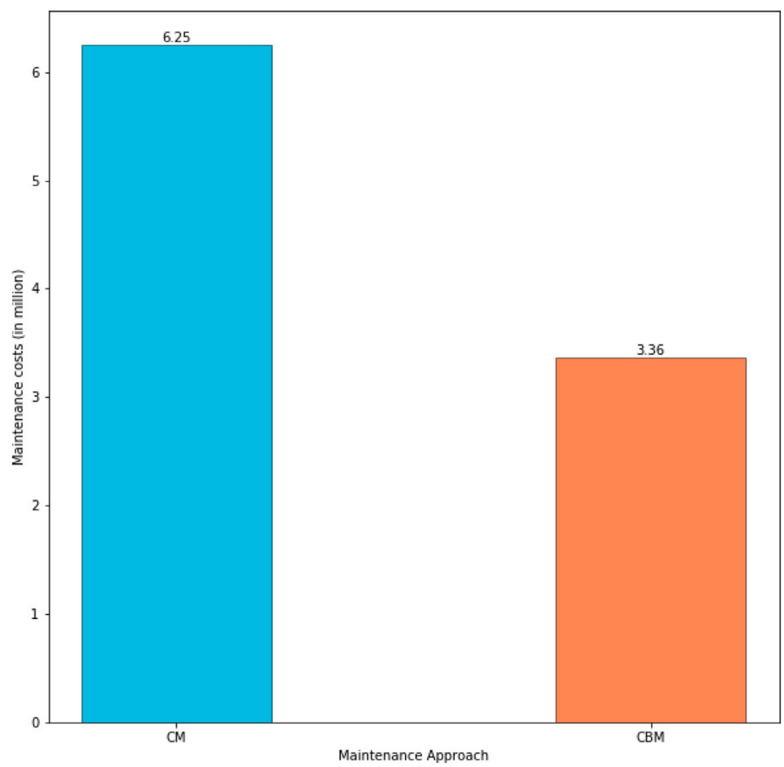

Fig. 12. Costs per maintenance approach.

having a total of 250 prognostics-driven tasks and 1517 preventive and corrective maintenance tasks. The results show that 96.4% of the considered monitored components were maintained on time. Moreover, the introduction of the prognostics-driven tasks can lead approximately to 46% reduction in maintenance costs. Besides that, the results show that the final output of our model reduces the used maintenance slots and the last-minute schedule changes. Overall, our approach produces stable and efficient maintenance schedules and is computationally efficient for quasi-real-time, while operating in an uncertain environment.

We believe this model to be the first of its kind in the context of aircraft CBM planning. As such it can be improved in many ways. In particular, future work should focus on evaluating the output of the model when using different types of prognostics models and distributions for capturing the RUL prognostics uncertainty. Moreover, additional maintenance actions, like inspections, can be incorporated into the formulation of the scheduling block for the prognostics-driven tasks. The additional actions can be assessed using metrics like the Value of Information (VoI). Finally, this research included only tasks that required execution in the hangar. However, in airline practice, many maintenance tasks are resolved in line maintenance. Extending the framework to include line maintenance capabilities can significantly increase schedule efficiency.

CRediT authorship contribution statement

Iordanis Tseremoglou: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bruno F. Santos:** Writing – review & editing, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This research was funded by the European Union's Horizon 2020 research and innovation program under the REMAP project, grant number 769288.

Appendix A. Algorithm for the derivation of maintenance policy for each monitored component for the intended planning horizon

Algorithm 1 Derivation of maintenance policy for each monitored component for the intended planning horizon

Set desired planning horizon
if new prediction/observation is received **then**
  Prune tree at the received observation/corresponding belief node b_t
  Discard all the other belief nodes
  Run POMCP algorithm as defined in [4]
  Select action $\hat{a}$ with the highest value function
  Append action $\hat{a}$ to the list of selected maintenance actions
  repeat()
    Prune tree at the belief node b_t with the highest amount of visits N
    Discard all other belief nodes
    Run POMCP algorithm as defined in [4]
    Select action $\hat{a}$ with the highest value function
    Append action to the list of selected maintenance actions
  until Terminal state or end of planning horizon
else
  repeat()
    Prune tree at the belief node b_t with the highest amount of visits N
    Discard all other belief nodes
    Run POMCP algorithm as defined in [4]
    Select action $\hat{a}$ with the highest value function
    Append action $\hat{a}$ to the list of maintenance actions
  until Terminal state or end of planning horizon
end if

Appendix B. Greedy algorithm-allocation of aircraft tasks to maintenance slots

Algorithm 2 Greedy algorithm-allocation of aircraft tasks to fixed maintenance slots

INPUT R, G, M_{fixed}
for $r \in R$ **do**
  repeat()
    Sort the aircraft tasks, $g \in G_r$, in ascending order according to the number of remaining maintenance opportunities $m_r \in M_{fixed}$, verifying by the set of constraints (2)–(7).
    Add the aircraft tasks that have no remaining maintenance opportunities to the list of tasks to be considered for scheduling by the DRL algorithm.
    Select the first element from the sorted list, i.e., the most urgent task g_{urgent}
    if $g_{urgent} \in G_{prev} \cup G_{progn}$ **then**
      Assign g_{urgent} to the latest possible fixed maintenance slot.
    else if $g_{urgent} \in G_{corr}$ **then**
      Assign g_{urgent} to the earliest possible fixed maintenance slot.
    end if
    Update the available workhours of the used maintenance slot
  until all aircraft tasks, $g \in G_r$, have been considered
end for

Appendix C. Definition of the reward function r_m

The first and most important objective is to keep the aircraft safe and airworthy, i.e., to prevent maintenance tasks from going due. For this reason, in lines 3–4, a large negative penalty, $W_{fail} \ll 0$, is returned when the number of due tasks increases between two successive maintenance slots. Also, one of the objectives is to improve maintenance schedule stability and subsequently, increase the flight schedule reliability. To account for this, in lines 5–6, a negative penalty $W_{fail} \ll W_{resc} \ll 0$ is introduced when a schedule change is observed within the period defined by the airline that schedule changes should be prevented. Moreover, as we move closer to the day of operations, the impact of a schedule change is higher, and therefore the introduced penalty increases linearly.

In lines 8–9 a positive reward, set equal to W_{ground}, is returned when the agent decides not to schedule any aircraft for maintenance, as in this case, the aircraft is available for operations.

Lines 11–12 ensure that the agent prefers to assign the aircraft to maintenance slots (if they exist) where all the open tasks can be addressed at once, hence minimizing the use of maintenance slots. In case only the current considered slot can accommodate all open maintenance tasks ($FS_r = 1$) and there are no forthcoming maintenance slots where all tasks can be addressed ($FNS_r = 0$), the highest possible reward is returned when the agent allocates the aircraft to this slot. In the opposite case, i.e., when $FS_r = 0$ and $FNS_r = 1$, a zero reward is returned when the agent assigns the aircraft to the current slot.

Finally, lines 14–18 guide the selection of the algorithm when either there are multiple maintenance slots that can accommodate all the aircraft open tasks, or there are no slots where all tasks can be addressed at once. In this case, the first objective that needs to be satisfied is to allocate the tasks to maintenance slots that have a similar duration to the duration of the considered tasks, i.e., a high DR_r is achieved. This is accomplished by the first term in the reward function, R_m, in lines 16 & 18, where the highest reward is returned when $DR_r = MDR_r$, i.e., when the algorithm selects the maintenance slot which, in

Algorithm 3 Definition of the reward R_m for every state-action pair (s_m, a_m) at every maintenance slot m

1: $W_{fail} << W_{resch} << 0$
2: $W_{ground} > W_{util} >> 0$
3: **if** $DT_{m+1} > DT_m$ **then** ▷ Case where task(s) go due
4: $\quad \mathbf{r_m} = \underbrace{W_{fail} \times (DT_{m+1} - DT_m)}_{\text{Task execution}}$
5: **else if** $SC_{m+1} > SC_m$ **then** ▷ Case where a schedule change is performed
6: $\quad \mathbf{r_m} = \underbrace{W_{resch} \times max(CP - ND_m, 0)}_{\text{Schedule stability}}$
7: **else**
8: $\quad$ **if** $a_m = NULL$ **then** ▷ Case where no aircraft is scheduled for maintenance
9: $\quad\quad \mathbf{r_m} = W_{ground}$
10: $\quad$ **else** ▷ Case where an aircraft is allocated to a maintenance slot
11: $\quad\quad$ **if** $FS_r \neq FNS_r$ **then**
12: $\quad\quad\quad \mathbf{r_m} = \underbrace{(W_{ground} + W_{util}) \times FS_r}_{\text{Aircraft operational availability}}$
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ **if** $FS_r = FNS_r$ **then**
15: $\quad\quad\quad\quad$ **if** $TT_r = 0$ **then**
16: $\quad\quad\quad\quad\quad \mathbf{r_m} = \underbrace{W_{ground} \times (1 - (MDR_r - DR_r))}_{\text{Aircraft operational availability}} + \underbrace{W_{util} \times (1 - (MRU_r - RU_r))}_{\text{Task interval utilization}}$
17: $\quad\quad\quad\quad$ **else**
18: $\quad\quad\quad\quad\quad \mathbf{r_m} = \underbrace{W_{ground} \times (1 - (MDR_r - DR_r))}_{\text{Aircraft operational availability}} + \underbrace{W_{util} \times (1 - NRU_r)}_{\text{Task interval utilization}}$
19: $\quad\quad\quad\quad$ **end if**
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end if**
22: $\quad$ **end if**
23: **end if**

terms of duration, is the best possible fit for the considered tasks. The last objective in terms of importance is the task interval utilization. In case there are preventive and prognostics-driven tasks ($TT_r = 0$), the second term in the reward function, R_m, in line 16, returns the highest reward when $MRU_r = RU_r$, i.e., when the algorithm selects the maintenance slot that provides the highest possible interval utilization rate. When there are only corrective maintenance tasks ($TT_r = 1$), the objective is to schedule them as soon as possible, which is achieved by the second term of the reward function, R_m, in line 18. This term returns a high reward when a low NRU_r is achieved, i.e. when the algorithm schedules the considered tasks as soon as possible. Because in the airline practice the objective of increased aircraft operational availability has a higher priority than the objective of task interval utilization, we set $W_{ground} > W_{util}$.

References

[1] Sprong J, Jiang X, Polinder H. Deployment of prognostics to optimize aircraft maintenance-A literature review. J Int Bus Res Mark 2020;5:26–37.
[2] Ingemarsdotter E, Kambanou M, Jamsin E, Sakao T, Balkenende R. Challenges and solutions in condition-based maintenance implementation - a multiple case study. J Clean Prod 2021;296.
[3] Meissner R, Rahn A, Wicke K. Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making. Reliab Eng Syst Saf 2021;214.
[4] Silver D, Veness J. Monte-Carlo planning in large POMDPs. In: Proc. Neural Inf. Process Syst. 2010, p. 1–9.
[5] Zhao X, Al-Khalifa KN, Magid Hamouda A, Nakagawa T. Age replacement models: A summary with new perspectives and methods. Reliab Eng Syst Saf 2017;161:95–105.
[6] Deng Q, Santos B, Curran R. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. European J Oper Res 2020;281:256–73.
[7] Andrande P, van Silva C, Ribeiro B, Santos B. Aircraft maintenance check scheduling using reinforcement learning. Aerospace 2021;8.
[8] Lagos C, Delgado F, Klapp M. Dynamic optimization for airline maintenance operations. Transp Sci 2020;54:998–1015.
[9] van Kessel PJ, Freeman FC, Santos BF. Airline maintenance task rescheduling in a disruptive environment. European J Oper Res 2022.
[10] Alaswad S, Xiang Y. A review on condition-based maintenance optimization models for stochastically deteriorating systems. Reliab Eng Syst Saf 2017;157.
[11] Powell W, Ryzhov I. Optimal learning. Wiley and Sons; 2012.
[12] Nguyen K, Do P, Huynh K, Berenguer C, Grall A. Joint optimization of monitoring quality and replacement decisions in condition-based maintenance. Reliab Eng Syst Saf 2019;189:177–95.
[13] Liu X, Sun Q, Ye Z-S, Yildirim M. Optimal multi-type inspection policy for systems with imperfect online monitoring. Reliab Eng Syst Saf 2021;207:107335.
[14] Song C, Zhang C, Shafieezadeh A, Xiao R. Value of information analysis in non-stationary stochastic decision environments: A reliability-assisted POMDP approach. Reliab Eng Syst Saf 2022;217:108034.
[15] Zhao Y, Smidts C. Reinforcement learning for adaptive maintenance policy optimization under imperfect knowledge of the system degradation model and partial observability of system states. Reliab Eng Syst Saf 2022;224:108541.
[16] Papakonstantinou K, Shinozuka M. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation. Reliab Eng Syst Saf 2014;130:214–24.
[17] Andriotis C, Papakonstantinou K. Managing engineering systems with large state and action spaces through deep reinforcement learning. Reliab Eng Syst Saf 2019;191.
[18] Andriotis C, Papakonstantinou K. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. Reliab Eng Syst Saf 2021;212.
[19] Zhang N, Si W. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. Reliab Eng Syst Saf 2020;203:107094.
[20] Mohammadi R, He Q. A deep reinforcement learning approach for rail renewal and maintenance planning. Reliab Eng Syst Saf 2022;225:108615.
[21] De Pater I, Mitici M. Predictive maintenance for multi-component systems of repairables with remaining-useful-life prognostics and a limited stock of spare components. Reliab Eng Syst Saf 2021;214.
[22] De Pater I, Reijns A, Mitici M. Alarm-based predictive maintenance planning for aircraft engines with imperfect RUL prognostics. Reliab Eng Syst Saf 2022;221.

[23] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic RUL prognostics. Reliab Eng Syst Saf 2023;230.
[24] Smallwood R, Sondik E. The optimal control of partially observable Markov processes over the infinite horizon: Discounted cots. Oper Res 1973;21:1071–88.
[25] Kocsis L, Szepesvari C. Bandit based Monte-Carlo planning. In: Proc. of European conference on machine learning. 2006, p. 282–93.
[26] Bieber M, Verhagen W, Santos B. An adaptive framework for remaining useful life predictions of aircraft systems. In: PHM Society European conference, vol. 6. 2021, p. 60–70.
[27] Saxena A, Goebel K. Turbofan engine degradation simulation dataset. NASA Ames Prognost Data Reposit 2008;878–87.
[28] Tseremoglou I, Bieber M, Santos B, Floris F, van Kessel P. The impact of prognostic uncertainty on condition-based maintenance scheduling: an integrated approach. In: AIAA aviation forum. American Institute of Aeronautics and Astronautics Inc.; 2022.
[29] Freeman F, van Kessel P, Verhagen W. Age and condition-based preventive replacement timing for periodic aircraft maintenance checks. In: Proceedings of the 6th European conference of the Prognostics and Health Management Society. 2021, p. 151–61.
[30] Tseremoglou I, van Kessel P, Santos B. A comparative study of optimization models for condition-based maintenance scheduling of an aircraft fleet. Aerospace 2023;10.