Lookahead Approximate Dynamic Programming for Stochastic Aircraft Maintenance Check Scheduling Optimization

Deng, Q.; Santos, Bruno F.

DOI
10.1016/j.ejor.2021.09.019

Publication date
2021

Document Version
Accepted author manuscript

Published in
European Journal of Operational Research

Lookahead Approximate Dynamic Programming for Stochastic Aircraft Maintenance Check Scheduling Optimization

Qichen Deng*, Bruno F. Santos

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

Abstract

This paper proposes a lookahead approximate dynamic programming methodology for aircraft maintenance check scheduling, considering the uncertainty of aircraft daily utilization and maintenance check elapsed time. It adopts a dynamic programming framework, using a hybrid lookahead scheduling policy. The hybrid lookahead scheduling policy makes the one-step optimal decision for heavy aircraft maintenance based on deterministic forecasts and then determines the light maintenance according to stochastic forecasts. The objective is to minimize the total wasted utilization interval between maintenance checks while reducing the need for additional maintenance slots. By achieving this goal, one is also reducing the number of maintenance checks and increasing aircraft availability while respecting airworthiness regulations. We validate the proposed methodology using the fleet maintenance data from a major European airline. The descriptive statistics of several test runs show that, when compared with the current practice, the proposed methodology potentially reduces the number of A-checks by 1.9%, the number of C-checks by 9.8%, and the number of additional slots by 78.3% over four years.

Keywords: Scheduling, Approximate Dynamic Programming, Lookahead Policy, Stochastic Optimization

1. Introduction

The aircraft maintenance check scheduling (AMCS) problem determines when and how often a type of maintenance check should be performed on an aircraft. AMCS for a large fleet and several check types is an intricate problem due to its combinatorial nature and real-life operational constraints. To ensure aircraft airworthiness, maintenance planners of airlines have to schedule maintenance inspections regularly for each aircraft before it reaches certain thresholds. These thresholds are in the units of calendar days (DY), flight hours (FH), or flight cycles (FC), stated in the maintenance planning document (MPD), as shown in Table 1. The maintenance planners allocate aircraft to maintenance slots on specific days, in which one maintenance slot is one day of availability of a hangar for performing aircraft maintenance. The maintenance schedule developed by maintenance planners is, however, subject to frequent disruptions. Weather conditions or flight disruption can impact aircraft utilization and further cause deviation from the original maintenance plan.

*Corresponding Author

Email address: q.deng@tudelft.nl (Qichen Deng)

Table 1: Definitions of aircraft usage parameters according to MPD.

Usage Parameter	Abbreviation	Description
Calendar Days	DY	A 24-hour period
Flight Hours	FH	The elapsed time between wheel lift off and touch down
Flight Cycles	FC	A complete take-off and landing sequence

Besides, the non-routine tasks/works affect the maintenance elapsed time and, therefore, the duration that an aircraft stays on the ground. These uncertainties make the AMCS problem challenging, as the maintenance planners have to regularly adapt the aircraft maintenance schedule. Following a manual or deterministic scheduling approach may result in insufficient hangar availability in specific moments, requiring the creation of extra maintenance slots, which are much more costly than regular maintenance slots.

In practice, maintenance planners of airlines usually grouped the aircraft maintenance tasks into letter checks depending on the level of detail: A-, B-, C-, and D-check, as shown in Table 2. A- and B-checks are considered light maintenance, and C- and D-check as heavy maintenance. Furthermore, C-/D-checks are more detailed inspection than A-/B-checks and require more maintenance resources (e.g., tools, workforce, and aircraft spare parts) and time to complete the maintenance tasks. Hence, C-/D-checks have higher priorities than A-/B-checks. In some cases, airlines can distribute the tasks within a B-check into successive A-checks or incorporate the items to be maintained in a D-check into multiple C-checks. We still adopt the classic letter check classification, and the AMCS optimization is equivalent to allocating A-/B-/C-/D-checks for the right aircraft at the right time.

Table 2: Aircraft letter check and corresponding inspection interval (Ackert, 2010).

Check	Maintenance Type	Interval	Maintenance Tasks
A-check	Light maintenance	2–3 months	External visual inspection, filter replacement, lubrication etc.
B-check	Light maintenance	Rarely mentioned	Tasks are commonly incorporated into successive A-checks
C-check	Heavy maintenance	18–24 months	Thorough inspection of the individual systems and components
D-check	Heavy maintenance	6–10 years	Thorough inspection of most structurally significant items

The current focus, in the literature and practice, has been primarily on the short-term maintenance planning, such as A-/B-check scheduling (Sriram and Haghani, 2003; Lagos et al., 2020), line maintenance planning (Papakostas et al., 2010; Shaukat et al., 2020), or coupled in the literature with the definition of the aircraft routing for the next three to six days of operations (Başdere and Bilge, 2014; Liang et al., 2015). However, one primary deficiency of a short-term horizon for aircraft maintenance planning is that it often neglects the importance of heavy maintenance scheduling. Optimizing short-term maintenance activities can result in a greedy policy deferring all letter checks to their due dates. If the maintenance planners defer a letter check, they may not see any maintenance capacity problem in the coming one or two months. Yet, the maintenance check overload can happen a year later. In other words, one may get a false impression that the maintenance resources meet the demands of letter checks in a short period, but, as time moves on, the following letter checks can pile up and cause soaring demand for maintenance in the future. Although some authors were aware of this issue in the 1970's (Boere, 1977), it was not straightforward to estimate the cost of current maintenance decisions for the future.

In 2015, the AIRMES project was launched by the European Commission to optimize end-to-end maintenance activities within an operator's environment (European Commission, 2015). One of the work packages within AIRMES has the mission of addressing the AMCS problem and minimizing the long-term aircraft maintenance costs for all maintenance check types. Aircraft maintenance is one of the main direct operating costs of an airline. In 2018, the spend of global maintenance, repair, and overhaul (MRO) represented 9%–10% of total operational costs, which was valued at $69 billion, excluding overhead (e.g., lighting, equipment, and any little extras), for a total number of 27.5K aircraft (IATA's Maintenance Cost Task Force, 2019). This spending was equivalent to $2.5M per aircraft per year. Base maintenance (including all check types) accounts for 20% of the $2.5M, excluding the cost for engine maintenance and components. An aircraft will be removed from the revenue schedule when it is undergoing maintenance, which could represent a loss of $75K–$120K of commercial revenue per day. Moreover, if airlines have to create additional maintenance slots, they have to spend more money to let the maintenance technicians work overtime or hire another company to perform the maintenance checks at a much higher cost. Therefore, airlines are laying increasing emphasis on improving their aircraft availability and planning their maintenance more carefully and efficiently.

Deng et al. (2020) proposed a solution to deterministic AMCS resulting from the AIRMES project. However, one of the limitations is that the optimization model described in Deng et al. (2020) assumes complete information and does not include future uncertainty. Despite other disruptions, such as flight delays, the maintenance schedule is affected by the elapsed time of maintenance checks. The stochastic AMCS has not been tackled so far, not even adequately studied. Since it is in general impossible for airlines to follow a long-term aircraft maintenance schedule without adjustment, maintenance planners have to update the maintenance schedules from time to time due to flight disruptions or changes in maintenance tasks execution.

For each aircraft letter check, the maintenance tasks are divided into two parts: routine maintenance tasks and non-routine maintenance tasks. For a specific check type, the routine maintenance tasks are the ones that are repeatedly scheduled and executed during the checks. The non-routine tasks include, e.g., replacement of major components (e.g., aircraft engines or landing gears), airworthiness directives (Transport Canada, 2008), engineering orders (Commercial Aviation Safety Team, 2013), deferred tasks, non-scheduled maintenance tasks that result from faults or additional maintenance needs found when executing the routine task. These non-routines can be up to 50% of the workload performed during a maintenance check (Alfares, 1999; Samaranayake and Kiridena, 2012). Most non-routine tasks are only known a few weeks or days before a maintenance check starts, and some during the aircraft maintenance check execution.

To cope with uncertainties and respond to changes in maintenance activities promptly, we propose a fast, short-term decision-making solution without comprising the long-term benefit. This work is the continuation of our previous maintenance planning optimization solution (Deng et al., 2020), extending the AMCS to a stochastic framework that considers uncertainty associated with aircraft utilization and maintenance check elapsed time. A lookahead approximate dynamic programming (ADP) methodology is presented and used, for the first time, to address the stochastic daily decisions for the AMCS. The contributions of this paper include:

- *Methodology*: The proposed hybrid policy of the lookahead ADP methodology is original and novel. It uses deterministic forecasts to estimate the number of extra maintenance slots in the future for heavy maintenance and stochastic forecasts to estimate the extra slots for frequent light maintenance.
- *Application*: The proposed methodology is more robust than the previous deterministic approach present in the literature, both in terms of fewer expected number of maintenance changes and additional maintenance slots.
- *Practicality*: It takes only seconds to determine the optimal maintenance check for the next day, significantly reducing the time needed for updating the letter check schedule. The proposed lookahead ADP methodology can help maintenance planners develop and adapt short-term maintenance check schedules within seconds without compromising the long-term efficiency of the solution.

The rest of the paper is divided into six sections. Following the introduction, Section 2 gives an overview of the literature about stochastic scheduling. Section 3 defines and formulates the stochastic AMCS problem. A lookahead ADP methodology is presented in Section 4, including the associated model framework and a hybrid lookahead policy. In Section 5, we show two case studies from a European airline. The last section concludes the paper and gives an outlook on future work.

2. Literature Review

Several publications address the aircraft maintenance related problems considering the stochastic elements. The earliest one can be traced back to 1966, in which Jorgenson et al. (1966) provided a unified view of maintenance from the theoretical perspective and its application on aircraft equipment. This technical report mainly focuses on the aircraft component level, and the primary source of uncertainty is the failure rate of aircraft equipment. The optimization model and associated solution techniques described are dedicated to individual aircraft systems or components. It is worth mentioning that the fleet size of airlines was much smaller back then since traveling by plane was expensive and dangerous in the 1960s (Brownlee, 2013), and the maintenance programs were process-oriented (SKYbrary, 2019).

Other than finding optimal maintenance policies for aircraft systems or components, some research works focus on minimizing the total time needed for aircraft maintenance activities considering uncertainties. Tsai and Gemmill (1998) applied tabu search on the coordination of aircraft maintenance activities to reduce the duration of all project activities, which was shown efficient for both deterministic and stochastic problems. The main idea behind the tabu search is to apply local search to improve an initial sequence of maintenance activities. But different from the classic tabu search, the authors introduced multiple tabu lists and randomized short-term memory to prevent solutions from being revisited, which significantly improved algorithm efficiency. Besides, multiple starting schedules were used to diversify local search to improve the optimality. To evaluate the performance of the tabu search, the authors compared the results from the tabu search and simulated annealing. The outcomes showed that tabu search outperformed simulated annealing in terms of a better aircraft maintenance schedule and shorter computation time.

Rosenberger et al. (2000) was aware that airline planning models did not explicitly consider stochastic elements in operations, which often led to discrepancies between the initial schedule and actual performance. To better capture the impact of uncertainty on daily airline operations (e.g., flight planning, crew paring, and maintenance scheduling), SimAir was developed to simulate and evaluate plans and recovery policies. SimAir consists of three modules: a random event generator to give random disruption, such as late arrival, ground time delay, or unscheduled maintenance delay; a recovery module to propose a recovery policy (revised schedule); a controller module to determine if a flight should be canceled due to disruption and whether or not a recovery policy should be accepted. The recovery module adopts a relatively trivial push-back strategy. For instance, if an unscheduled maintenance event causes a flight delay, the departure time of the flight will be deferred until the unscheduled maintenance tasks are finished. Although there were not many optimization techniques involved in this study, Rosenberger et al. (2000) still provides insights into how random disruptions affect the daily operation of airlines and how airlines recover from disruptions, which also prompts us to develop a dynamic optimal decision-making model for AMCS.

As mentioned in Rosenberger et al. (2000), stochastic simulation is a way of capturing uncertainty, particularly very essential in aircraft maintenance operations. The reason is straightforward: aircraft system or component failure appears to be random, and the maintenance activities are tightly coupled with each other in a sequence. Any delay in executing a task can have snowball effects on the following maintenance activities, which may eventually lead to a maintenance delay. Gupta et al. (2003) applied stochastic modeling and simulation on aircraft line maintenance (maintenance near the gate or terminal between aircraft arrival and departure) to investigate the potential of improving maintenance management. This research aimed at minimizing the total number of technicians working overtime under the uncertainty of maintenance activities. The authors applied a genetic algorithm to address the problem. The results from stochastic optimization indicated that the workload was likely to be better spread across shifts.

Aircraft maintenance operations are often plagued by planning difficulties because of maintenance activities and flight arrival. Maintenance delay or bad weather often results in late departure and, in the end, late arrival of a flight. Some airlines have been trying to plan a robust aircraft maintenance schedule or maintenance personnel rosters in the past few years. For example, Bruecker et al. (2015) proposed a model enhancement (ME) algorithm for planning robust aircraft maintenance personnel rosters to cope with stochastic flight arrival. The optimal aircraft maintenance personnel rosters minimize the total labor costs while achieving a certain service level. The main idea was to use stochastic simulation to simulate the flight arrivals and allocation of maintenance capacity to flights for several weeks. And this helps airlines to identify the flights that often cannot be maintained in time. Based on the simulation results, the algorithm adjusted workforce configuration by adding workforce to reduce the average number of flights that cannot be maintained; after that, a mixed-integer programming model was formulated and addressed by commercial solver CPLEX. The proposed algorithm was tested using the data from Sabena Technics (an aircraft maintenance company located at Brussels Airport). It was demonstrated to provide robust solutions. Following this idea, we use simulation to simulate aircraft utilization and maintenance elapsed time in this research, which gives us an estimation of when an aircraft needs to be maintained and how long a maintenance check lasts.

Several other studies about operational aircraft maintenance can be found in Papakostas et al. (2010), Eltoukhy et al. (2017), Eltoukhy et al. (2018) and Lagos et al. (2020), yet none of them deal with AMCS. The main reason is that AMCS involves both long-term (e.g., C-/D-check) and short-term planning (e.g., A-/B-check), and the goal of short-term planning may contradict the long-term objective. For instance, one common goal for short-term maintenance planning is to minimize the cost (Moudani and Mora-Camino, 2000; Sriram and Haghani, 2003). To achieve this goal, airlines tend to defer replacing components as close to their estimated due dates as possible, leading to grounding an aircraft too often and lower aircraft availability and thus lower revenue; or defer maintenance checks to their estimated due dates, which can result in capacity issues in the long term and much higher costs for creating extra maintenance capacity. On the other hand, the long-term goal is often to maximize profits. However, due to data availability, it is difficult to calculate the long-term labor and material costs or the long-term revenues generated from commercial operations. Therefore, it is necessary to have an objective that suits both short-term and long-term planning. Without revenue or maintenance cost data, minimizing total unused FH (Boere, 1977; Başdere and Bilge, 2014) is a good objective to unify the planning of all A-, B-, C- and D-checks since minimizing the total unused FH can also reduce the number of letter checks and maximize aircraft availability for commercial operations.

In theory, AMCS is close to the resource-constrained project scheduling problem (RCPSP), e.g., RCPSP has resource constraints and uncertain task duration (Li and Womer, 2015), and AMCS has maintenance capacity constraints and uncertain maintenance check elapsed time. The main difference is that AMCS also has uncertainty in aircraft utilization, which impacts the due dates of the following maintenance checks. Hence, the start/due date of a maintenance check depends on its previous execution and the utilization of the aircraft. Li and Womer (2015) proposed an approximate dynamic programming (ADP) approach for RCPSP based on a lookahead policy, combining a priority-rule heuristic for reducing problem dimensions and a lookup table for improving optimality. It was shown to perform well on 120 tasks in case studies. Although Li and Womer (2015) has small problem sizes in its case studies and the dimension of AMCS is too high to create a lookup table, it provides us inspiration and insights for developing an efficient lookahead policy for AMCS.

Based on our findings during the literature review, we draw the following conclusions. First of all, many papers propose robust short-term operational aircraft maintenance plans, recovery policies, or maintenance personnel rosters to cope with uncertainty. However, to our best knowledge, there is no literature found about AMCS optimization except for Deng et al. (2020). Secondly, stochastic simulation is a useful method to predict incidents (e.g., system failure, unscheduled maintenance, or flight delay). The simulation outcomes can provide insights about uncertainty and help maintenance planners make better aircraft maintenance check decisions. Lastly, even if one manages to find the optimal letter check schedule, it will most likely fail during real-life operations because of the rapid changing of aircraft utilization and maintenance environments, which requires lots of time or effort to recreate a new schedule. Since maintenance planners may need to update the letter check daily, it would be desirable to have a stochastic AMCS model to provide the optimal letter check decision every 24 hours according to the actual fleet utilization.

3. Problem Formulation

This paper adopts the same definitions and assumptions presented in Deng et al. (2020) to formulate the stochastic AMCS (S-AMCS) problem. The nomenclature and corresponding definition can be found in Appendix A. In essence, the S-AMCS is a typical Markov Decision Process (MDP) consisting of:

- A set of *decision epochs* $\{t_0, t_0+1, \ldots, T\}$
- A set of states $\{s_t\}$ called the *state space*
- A set of associated actions from a state s_t, $X_t(s_t) = \{\mathscr{X}^\pi(s_t)\}$, called the *action space*, where $\mathscr{X}^\pi(s_t)$ is the scheduling policy function
- The *immediate reward/cost* of doing an action, $C_t(s_t, x_t)$, where s_t is the state and x_t is the action
- The *transition probability* $p(s_{t+1}|s_t)$ of changing a state s_t to a new state s_{t+1}

State Space

We define the state vector s_t as a set of attributes that influence our decisions and this set also includes available maintenance slots of each check type:

$$s_t = \{A_t, M_t, N_t\}, \; A_t = \{a_{t,i}\}_{i=1}^{N_t}, \; M_t = \left\{M_t^k \middle| k \in K\right\}, \; K = \{\text{A-check}, \text{B-check}, \text{C-check}, \text{D-check}\}. \tag{1}$$

where A_t is a collection of attributes from all aircraft, M_t is the set of hangar capacity of all check types, and the capacity of each check type is denoted by M_t^k. N_t is the fleet size that may change during the planning horizon. $a_{t,i} = \left\{a_{t,i}^k \middle| k \in K\right\}$ and $a_{t,i}^k$, contains the attributes of aircraft i on t, with respect to check type k:

$$a_{t,i}^k = \{\underbrace{z_{t,i}^k(\omega_t), \delta_{t,i}^k, \eta_{t,i}^k,}_{\text{Type 1 }\left(a_{t,i}^{(1),k}\right)} \underbrace{\text{DY}_{t,i}^k, \text{FH}_{t,i}^k, \text{FC}_{t,i}^k, y_{t,i}^k,}_{\text{Type 2 }\left(a_{t,i}^{(2),k}\right)} \underbrace{L_i\left(y_{t,i}^k\right), \text{fh}_{t,i}, \text{fc}_{t,i}, \Delta L_i^\omega\left(y_{t,i}^k\right), \Delta\text{fh}_{t+1,i}^\omega, \Delta\text{fc}_{t+1,i}^\omega}_{\text{Type 3 }\left(a_{t,i}^{(3),k}\right)}\} \tag{2}$$

These attributes can be divided into three types:

- **Type 1** $a_{t,i}^{(1),k}$: Attributes at time t that impact the action x_t and are modified when there is new information or after a maintenance check starts, including $z_{t,i}^k(\omega_t)$, the actual end date of type k check of aircraft i computed on day t; $\delta_{t,i}^k$, a binary variable to indicate if aircraft i is undergoing type k check on day t; $\eta_{t,i}^k$, a binary variable to indicate if aircraft i needs an extra slot of type k check on day t. Here ω_t is the information arriving on t.

- **Type 2** $a_{t,i}^{(2),k}$: Attributes at time t that are updated every time based on their value at time $t-1$, including $\mathrm{DY}_{t,i}^k$, $\mathrm{FH}_{t,i}^k$ and $\mathrm{FC}_{t,i}^k$, the utilization parameters (DY, FH, and FC) of aircraft i of type k check on day t; $y_{t,i}^k$, next maintenance label for type k check of aircraft i on day t.

- **Type 3** $a_{t,i}^{(3),k}$: Attributes at time t that depend on exogenous information and can be estimated according to historical aircraft utilization and maintenance data, including $L_i\left(y_{t,i}^k\right)$, mean estimated elapsed time of next check with label $y_{t,i}^k$ of aircraft i; $\mathrm{fh}_{t,i}$ and $\mathrm{fc}_{t,i}$, the average daily fight hours and flight cycles of aircraft i at day t; $\Delta L_i^\omega\left(y_{t,i}^k\right)$, $\Delta\mathrm{fh}_{t+1,i}^\omega$ and $\Delta\mathrm{fc}_{t+1,i}^\omega$, the uncertainties of maintenance check elapsed time, daily flight hours and daily flight cycles, respectively.

The uncertainties come from the attributes of **Type 3**, the aircraft utilization, and maintenance check elapsed time. For aircraft utilization, maintenance planners only obtain the exact aircraft FH and FC at the end of the day. For the actual maintenance check elapsed time, it is only known when a letter check starts. The update of each type of attributes is presented later in Subsection 3.2.

Action Space

In S-AMCS, the action space associated with a state s_t is a set of maintenance check actions. An action x_t of the day t is to perform one or several maintenance checks or do nothing:

$$x_t = \left\{ \left\{\chi_{t,i}^k\right\}_{i=1}^{N_t} \middle| \sum_{i=1}^{N_t} \chi_{t,i}^k \leq M_t^k \right\}_{k\in K} \tag{3}$$

where, each $\chi_{t,i}^k$ is a binary decision variable in which:

$$\chi_{t,i}^k = \begin{cases} 1 & \text{a type } k \text{ check for aircraft } i \text{ is planned to start at time } t \\ 0 & \text{otherwise (including the case that aircraft } i \text{ is undergoing a type } k \text{ check)} \end{cases} \tag{4}$$

For example, given 3 aircraft, there is 1 slot for A-check and 1 slot for C-check (1 slot is 1 day of availability of a hangar), and A-check can be merged in C-check, there are 25 actions, as shown in Table 3.

Immediate Reward

For an aircraft i, the value of unused FH of type k check on a day t is equal to the summation of the FH loss due to an A-/B-/C-/D-check scheduled for that day:

$$C_{t,i} = \sum_{k\in K} \chi_{t,i}^k \left(I_{k\text{-FH}}^i - \mathrm{FH}_{t,i}^k\right) \tag{5}$$

Table 3: An example of action space for three aircraft AC-1, AC-2 and AC-3, one A-check slot and one C-check slot. A-check can be merged in C-check without using an A-check slot. $\{\{\text{AC-}i, \text{AC-}j\}, \text{AC-}j\}$ means performing A-check on aircraft i, and both A- and C-check of aircraft j.

Action	**Aircraft Selection**	**Total Possibilities**
0 A-check and 0 C-check	—	1
0 A-check and 1 C-check	{—, AC-1}, {—, AC-2}, {—, AC-3}	3
1 A-check and 0 C-check	{AC-1, —}, {AC-2, —}, {AC-3, —}	3
1 A-check and 1 C-check	{AC-1, AC-1}, {AC-1, AC-2}, {AC-1, AC-3} {AC-2, AC-1}, {AC-2, AC-2}, {AC-2, AC-3} {AC-3, AC-1}, {AC-3, AC-2}, {AC-3, AC-3}	9
2 A-check and 1 C-check	{{AC-2, AC-1}, AC-1}, {{AC-3, AC-1}, AC-1} {{AC-1, AC-2}, AC-2}, {{AC-3, AC-2}, AC-2} {{AC-1, AC-3}, AC-3}, {{AC-2, AC-3}, AC-3}	9

where $\chi_{t,i}^{k}$ is a binary variable to indicate if aircraft i starts a type k check on t, and $I_{k\text{-FH}}^{i}$ is the interval of type k check of aircraft i in terms of FH. The *immediate reward*, or so called *contribution function*, is:

$$C_t\left(s_t, x_t\right) = \sum_{i=1}^{N_t} \left(C_{t,i} + \lambda \sum_{k \in K} \eta_{t,i}^{k} \right) \tag{6}$$

The first term on the right-hand side of (6) reflects the unused FH of aircraft i, the second term is a penalty for creating an additional slot of type k check on the day t. The penalty λ is introduced because creating one extra slot is equivalent to hiring a group of technicians to perform a maintenance check on extra work-hours on the day t or subcontracting the maintenance check to a third party MRO. This action is costly, and it should only be an option if it avoids an aircraft on the ground waiting for a maintenance slot. Hence, the value of λ should be much larger than $C_{t,i}$.

Transition Probability

The transition probability indicates the possibility of changing a state s_t to a new state depending on the new information ω_{t+1}. Here we use Figure 1 to illustrate MDP and state transition from stage t_0 to stage t_0+1. In this example, s_{t_0} is the initial state and $\{x_{t_0,j}\}$ is the set of associated actions. After making a decision $x_{t_0,j}$, we move from s_{t_0} to $\hat{s}_{t_0,j}$ but the new information ω_{t_0+1} has not arrived yet. ω_{t_0+1} is a stochastic variable arriving on t_0+1, each realization of ω_{t_0+1}, $\omega_{t_0+1}^{l}$, is associated with a transition probability $p_{t_0+1,l}$, meaning that ω_{t_0+1} has a probability $p_{t_0+1,l}$ of becoming $\omega_{t_0+1}^{l}$. As a result, $\hat{s}_{t_0,j}$ has a probability $p_{t_0+1,l}$ of becoming $s_{t_0+1,j}^{l}$. To facilitate understanding of the S-AMCS model, this section first describes the objective function in Subsection 3.1, then the state transition in Subsection 3.2, and the constraints in Subsection 3.3.

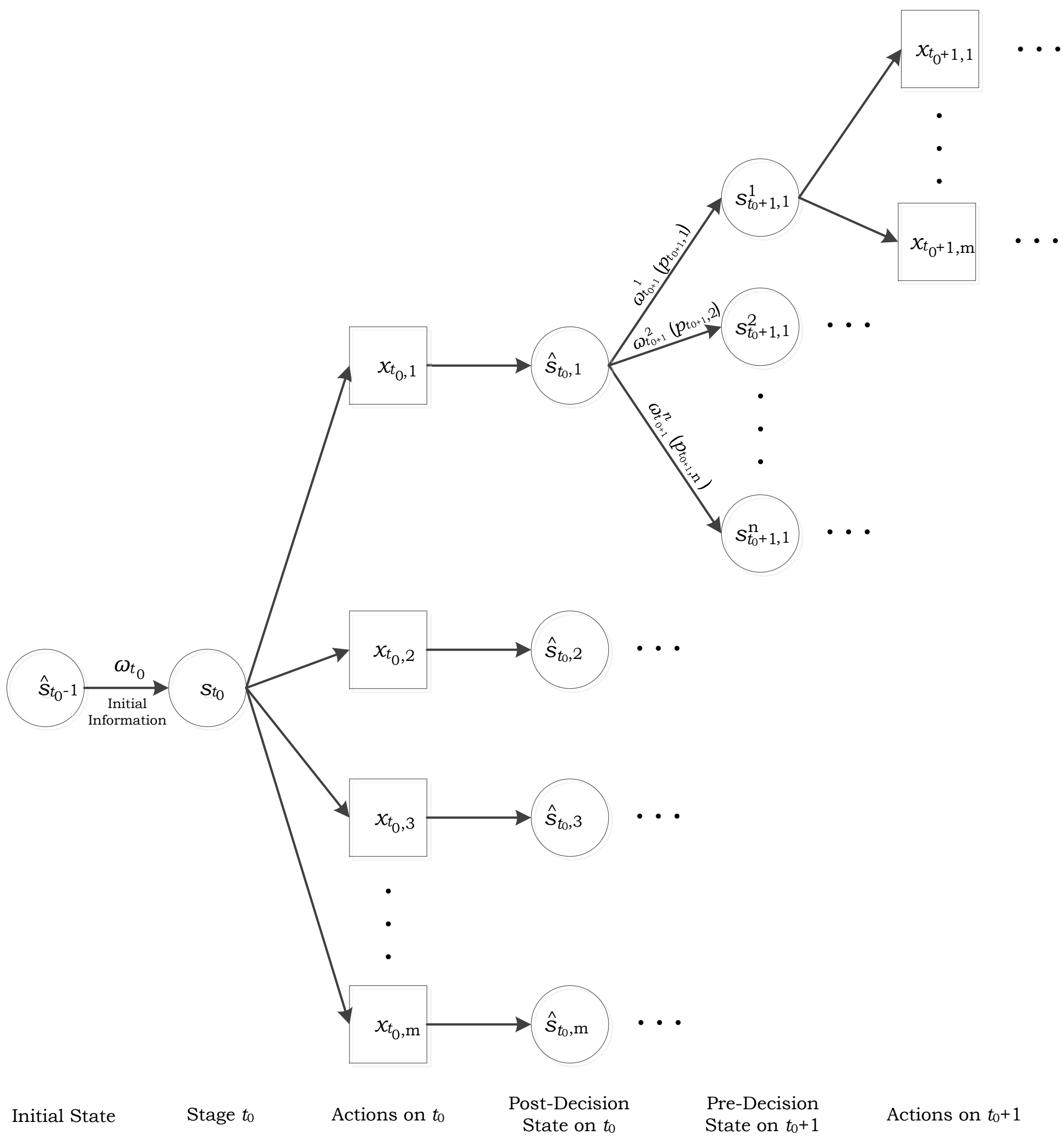


Figure 1: An example of state transition from stage t_0 to stage $t_0 + 1$ in stochastic AMCS. $\{x_{t_0,j}\}$ is the set of possible actions associated with s_{t_0} and $\{\hat{s}_{t_0,j}\}$ is the set of resulting post-decision states. The pre-decision state $s^j_{t_0+1,1}$ is only known when new information $\omega^j_{t_0+1}$ arrives and $p_{t_0+1,j}$ is the probability of transitioning the state $\hat{s}_{t_0,1}$ to $s^j_{t_0+1,1}$.

3.1. Objective Function

In the AMCS problem, the most common objectives are minimizing total maintenance costs (Moudani and Mora-Camino, 2000; Sriram and Haghani, 2003), and minimizing the total unused FH of a fleet (Boere, 1977; Başdere and Bilge, 2014; Deng et al., 2020). In this work, we chose the latter, i.e., minimizing the total unused FH, as in Deng et al. (2020) because of the following reasons:

- The available cost data is unreliable and hard to associate with a specific maintenance check;
- Maintenance checks are mandatory, and the total maintenance costs of an airline can only be reduced if the number of aircraft checks over time is also reduced;
- One day of an aircraft out of operations is more costly than the daily cost of a maintenance check.

Our objective is to minimize the sum of the total contributions for all states visited during the time horizon, discounted by a factor γ. That is, we search for the optimal AMCS policy (π) that minimizes the contribution of our scheduling decisions over the time horizon $[t_0, T]$:

$$\min_{\pi} \ \mathbb{E}\left\{ \sum_{t=t_0}^{T} \gamma^{t-t_0} C_t\left(s_t, \mathscr{X}^{\pi}\left(s_t\right)\right) \middle| s_{t_0} \right\} \tag{7}$$

where π is the scheduling policy that generates actions based on s_t, s_{t_0} denotes the initial state. $\mathscr{X}^{\pi}\left(s_t\right)$ maps the state s_t to an action under policy π, and $C_t\left(s_t, \mathscr{X}^{\pi}\left(s_t\right)\right)$ refers to (6). The optimal S-AMCS policy can be found by recursively computing the Bellman's equation:

$$V_t(s_t) = \min_{x_t} \left\{ C_t\left(s_t, x_t\right) + \gamma \sum_{s_{t+1}} p\left(s_{t+1} \middle| s_t\right) V_{t+1}(s_{t+1}) \right\} \tag{8}$$

where $p\left(s_{t+1} \middle| s_t\right)$ is the probability of transitioning from state s_t to state s_{t+1}. Eq. (8) expresses the value of being at state s_t, by considering the immediate contribution of an action x_t and the future value $V_{t+1}(s_{t+1})$.

3.2. State Transition in Stochastic Aircraft Maintenance Check Scheduling

The main difference in model formulation between AMCS presented in Deng et al. (2020) and S-AMCS in this paper is that S-AMCS has a two-phase state transition, post-decision update after performing an action, and pre-decision update after revealing new information. As a result, there are two state vectors associated to the update, post-decision state vector $\hat{s}_t$ and pre-decision state vector s_t. The post-decision state vector before the arrival of new information is notated and defined as:

$$\hat{s}_t = \mathcal{S}^X\left(s_t, x_t\right) \tag{9}$$

where $\mathcal{S}^X$ denotes the state transition function without knowing any new information. In S-AMCS, we assume that the new information $\{\omega_t\}_{t=t_0+1}^{T+1}$ is revealed when a maintenance check starts, or an aircraft ends its daily operation, then we update the state vector:

$$s_{t+1} = \mathcal{S}^W\left(\hat{s}_t, \omega_{t+1}\right) \tag{10}$$

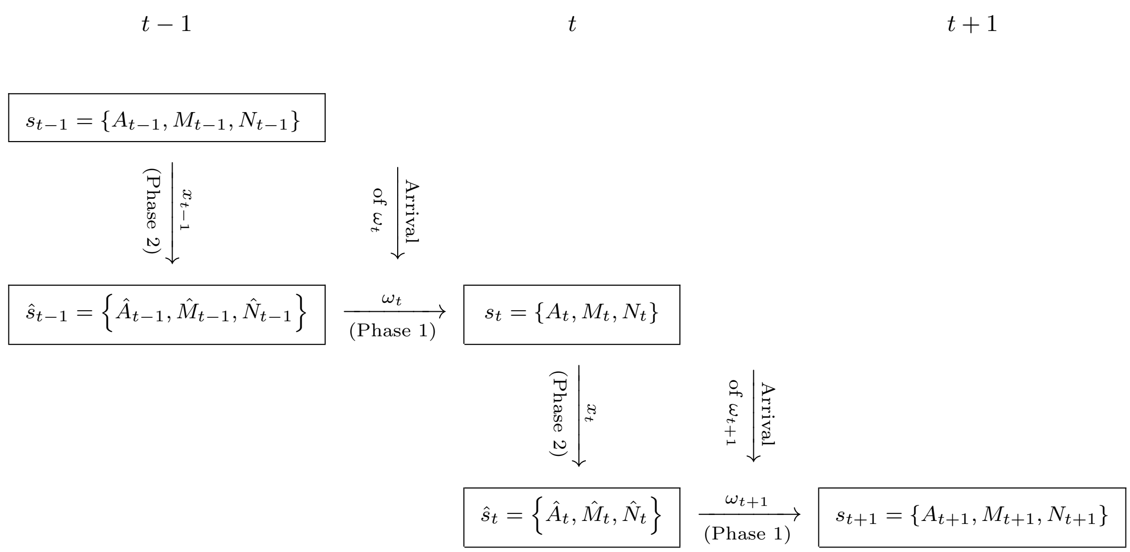


Figure 2: A two-phase attribute update mechanism: Phase 1 updates $\hat{s}_{t-1}$ to s_t before doing any action; after performing an action x_t, Phase 2 updates s_t to $\hat{s}_t$. ω_t includes the information of actual maintenance check elapsed time and actual aircraft daily utilization. $A_t = \{a_{t,1}, \ldots, a_{t,N_t}\}$, $M_t = \{M_t^k | k \in K\}$, $\hat{M}_t = \left\{\hat{M}_t^k | k \in K\right\}$ and $\hat{M}_t^k = M_t^k - \sum_{i=1}^{N_t} \hat{\delta}_{t,i}^k$.

where $\mathcal{S}^W$ is the transition function updating $\hat{s}_t$ to s_{t+1} according to the actual maintenance check elapsed time, daily FH and FC. A history of such a process, including the sequence of actions and evolution of states, can be represented as:

$$(\hat{s}_{t_0-1}, \omega_{t_0}, s_{t_0}, x_{t_0}, \hat{s}_{t_0}, \omega_{t_0+1}, s_{t_0+1}, x_{t_0+1}, ..., s_{t-1}, x_{t-1}, \hat{s}_{t-1}, \omega_t, s_t, ..., s_T, x_T, \hat{s}_T, \omega_{T+1}, s_{T+1}, ...) \tag{11}$$

The reason for including the post-decision state $\hat{s}_{t_0-1}$ as the initial state and initial information ω_{t_0} in (11) is that some aircraft might be undergoing maintenance checks in the initial state, ω_{t_0} is equivalent to knowing when those initial ongoing maintenance checks will be completed on the day t_0. The state transition from t to $t+1$ can be summarized in the following equations:

$$\begin{cases} s_t = \mathcal{S}^W(\hat{s}_{t-1}, \omega_t) \\ x_t = \mathscr{X}^\pi(s_t) \qquad \text{for} \quad t = t_0,\ t_0+1,\ ...,\ T. \\ \hat{s}_t = \mathcal{S}^X(s_t, x_t) \end{cases} \tag{12}$$

As shown in Figure 2, the state transition updates the attributes over the time horizon in two phases: pre-decision phase (Phase 1) updates the set of post-decision attributes $\hat{s}_{t-1}$ to s_t before performing any action, and post-decision (Phase 2) updates s_t to $\hat{s}_t$ after performing an action x_t. The new information, ω_t, arrives at the beginning of day t. According to the new information ω_t, the pre-decision phase (before making any new decision) renews the hangar capacity, aircraft availability, maintenance check elapsed time, and aircraft utilization based on actual FH and FC. This indicates, e.g., how many hangars can be used to perform maintenance checks on the day t, which aircraft is available for operation, and when an ongoing maintenance check will finish. In the post-decision phase, we update the aircraft usage parameters of each check type according to its actual daily utilization, and we also update the hangar occupation according to actual maintenance check elapsed time. Since we divide attributes of a state into three types, the transition of each type is presented separately in the following sub-sections.

3.2.1. Update of Type 1 Attributes

Since the actual elapsed time is only known when the check starts, namely, the new information arrives at t, in Phase 1, i.e., the pre-decision phase in Figure 2, we first check if t is the end day for an ongoing check before any action, or give the actual end date of a type k check if it starts at $t-1$ for all aircraft:

$$z^k_{t,i}(\omega_t) = \begin{cases} 0 & \text{if } \hat{z}^k_{t-1,i} = t-1 \\ \hat{z}^k_{t-1,i} + \Delta L^\omega_i(y^k_{t-1,i}) & \text{if } \chi^k_{t-1,i} = 1 \\ \hat{z}^k_{t-1,i} & \text{otherwise} \end{cases} \tag{13}$$

If the end date of a type k check of an aircraft i is $t-1$, $z^k_{t,i}(\omega_t)$ is set to 0. If the check just starts on $t-1$, $z^k_{t,i}(\omega_t)$ is updated by the expected end date $\hat{z}^k_{t-1,i}(\omega_t)$ plus the extra time $\Delta L^\omega_i\left(y^k_{t-1,i}\right)$, where $\Delta L^\omega_i\left(y^k_{t-1,i}\right)$ follows a certain distribution and its value depends on the realization ω_t. If the check started at least two days ago and the end date is larger than the current calendar day t, it means the check is still ongoing and $z^k_{t,i}(\omega_t)$ has the same value as $\hat{z}^k_{t-1,i}(\omega_t)$. According to the status of aircraft i, we update $\hat{\delta}^k_{t-1,i}$ to $\delta^k_{t,i}$:

$$\delta^k_{t,i} = \begin{cases} 0 & \text{if } z^k_{t,i} = 0 \\ \hat{\delta}^k_{t-1,i} & \text{otherwise} \end{cases} \tag{14}$$

The hangar capacity (available maintenance slots) also needs to be updated for time t accordingly:

$$M^k_t = \sum_{h_k} M^k_{t,h_k} - \sum_{i=1}^{N_t} \delta^k_{t,i} \tag{15}$$

where M^k_{t,h_k} is the maintenance capacity of a hangar h_k specifically for type k check at time t. The number of additional slots of type k check, $\eta^k_{t,i}$, is updated according to $\chi^k_{t,i}$:

$$\eta^k_{t,i} = \begin{cases} 0 & \text{if } \exists h_k,\ \sum_{\tau=t}^{t+L_i\left(y^k_{t,i}\right)} M^k_{\tau,h_k} - \chi^k_{t,i} L_i\left(y^k_{t,i}\right) \geq 0 \\ \min_{h_k} \left\{ \chi^k_{t,i} L_i\left(y^k_{t,i}\right) - \sum_{\tau=t}^{t+L_i\left(y^k_{t,i}\right)} M^k_{\tau,h_k} \right\} & \text{otherwise} \end{cases} \tag{16}$$

It is worth mentioning that we use a generic index h_k to represent a hangar in this paper. If one wants to consider multiple locations of performing the aircraft maintenance check, each hangar h_k would have to be associated with a location l_h and the decision variable $\delta^k_{t,i}$ will be replaced by $\delta^{l_h,k}_{t,i}$.

In Phase 2 (post-decision phase in Figure 2), the action x_t is taken into account to update Type 1 attributes. For all aircraft that start type k check on day t ($\chi^k_{t,i} = 1$), the values of $z^k_{t,i}$ and $\delta^k_{t,i}$ need to be updated. The $z^k_{t,i}$ is updated according to:

$$\hat{z}^k_{t,i} = \begin{cases} t + L_i\left(y^k_{t,i}\right) & \text{if } \chi^k_{t,i} = 1 \\ z_{t,i} & \text{otherwise} \end{cases} \tag{17}$$

Note that $L_i\left(y_{t,i}^k\right)$ refers to the mean elapsed time according to historical maintenance check data. The mean elapsed time is used in this study since no sufficient data points were available to statistically infer reliable maintenance elapsed time predictions. Following this update, the values of $\delta_{t,i}^k$ can also be renewed:

$$\hat{\delta}_{t,i}^k = \begin{cases} 1 & \text{if } \chi_{t,i}^k = 1 \\ \delta_{t,i}^k & \text{otherwise} \end{cases} \tag{18}$$

3.2.2. Update of Type 2 Attributes

Once the decision of the day t is known, the update of Type 2 attributes is trivial. The aircraft usage parameters are updated according to the following equations:

$$\mathrm{DY}_{t+1,i}^k = \left(1 - \hat{\delta}_{t,i}^k\right)\left(\mathrm{DY}_{t,i}^k + 1\right) \tag{19}$$

And the aircraft FH and FC are renewed according to new information ω_t:

$$\Psi_{t+1,i}^k = \left(1 - \hat{\delta}_{t,i}^k\right)\left(\Psi_{t,i}^k + \left(1 - \max_{k' \neq k}\left\{\hat{\delta}_{t,i}^{k'}\right\}\right)\left[\psi_{t,i} + \Delta\psi_{t+1,i}^{\omega}\right]\right), \quad \Psi \in \{\mathrm{FH}, \mathrm{FC}\}, \ \psi \in \{\mathrm{fh}, \mathrm{fc}\}. \tag{20}$$

where k' refers to the check type that is different from k, if k = A-check, k' can be any other check type (B-/C-/D-check) except for A-check. The usage parameters are reset to 0 if a maintenance check of type k is scheduled on the day t (i.e., $\hat{\delta}_{t,i}^k = 1$). Otherwise, the parameters are either increased by the average daily aging of the aircraft or remain the same, if a maintenance of the type other than k is scheduled (i.e., $\hat{\delta}_{t,i}^{k'} = 1$). $\Delta\psi_{t+1,i}^{\omega}$ follows a normal distribution and $\psi_{t,i}$ is the mean daily utilization of aircraft i according to airline estimation. Note that the decision variables $\chi_{t,i}^k$ do not explicitly impact the update of Type 2 attributes. $\chi_{t,i}^k$ are used to update Type 1 attributes directly, as shown in (16)–(18). $\hat{\delta}_{t,i}^k$ and $\hat{\delta}_{t,i}^{k'}$ are part of the results of Type 1 attributes update using decision variables $\chi_{t,i}^k$. Overall, $\chi_{t,i}^k$ and $\delta_{t,i}^k$ function differently in the S-AMCS model. We use $\chi_{t,i}^k$ to indicate the action related to type k check on aircraft i on day t, and $\delta_{t,i}^k$ to indicate whether aircraft i is undergoing a type k check or in commercial operations on day t.

After an action is determined, the maintenance labels for both type k checks are updated consequently. The maintenance labels of an aircraft i are updated to the next label using the following equation:

$$y_{t+1,i}^k = \begin{cases} y_{t,i}^k + 1 & \text{if } \chi_{t,i}^k = 1 \\ y_{t,i}^k & \text{otherwise} \end{cases} \tag{21}$$

3.2.3. Update of Type 3 Attributes

The Type 3 attributes are exogenous variables that are updated according to lookup tables, or provided by an airline, or estimated according to historical data of airline. They refer to:

- $L_i\left(y_{t,i}^k\right)$ is the mean elapsed time from historical maintenance data.
- $\mathrm{fh}_{t,i}$ and $\mathrm{fc}_{t,i}$ are estimated according to historical daily aircraft FH and FC.
- $\Delta L_i^{\omega}\left(y_{t,i}^k\right)$ follows an empricial distribution. $\Delta\mathrm{fh}_{t+1,i}^{\omega}$ and $\Delta\mathrm{fc}_{t+1,i}^{\omega}$ follow normal distributions. Their values all depend on the realization of ω_{t+1}. The new information ω_{t+1} arrives on day $t+1$.

Besides, we obtain M_{t+1} from the input data from airlines. We also have $\hat{N}_t = N_t$. $\hat{N}_t$ is updated to N_{t+1} according to the lifespan and utilization of each aircraft.

3.3. Constraints Formulation

There are two types of constraints in the AMCS optimization: maintenance check intervals and operational constraints. The maintenance checks are usually scheduled before the corresponding usage parameters reach maximums. This can be described as follows, for each check k, aircraft i, and time t:

$$\mathrm{DY}^k_{t,i} + 1 \leq I^i_{k\text{-DY}} \tag{22}$$

$$\Psi^k_{i,t} + \psi_{i,t} \leq I^i_{k\text{-}\Psi} \tag{23}$$

where $\Psi \in \{\mathrm{FH},\ \mathrm{FC}\}$ and $\psi \in \{\mathrm{fh},\ \mathrm{fc}\}$. This assessment is made on day t based on the mean daily FH and FC, before any new information arrives. If an aircraft reaches its maximum utilization but there is no maintenance slot available, an additional slot will be created to cope with extra maintenance demand.

The next constraint verifies whether or not there are sufficient maintenance slots for a type k check in all hangars during the entire mean maintenance elapse time $L_i\left(y^k_{t,i}\right)$:

$$\chi^k_{t,i} \leq \frac{\sum_{\tau=t}^{t+L_i\left(y^k_{t,i}\right)} M^k_{\tau,h_k}}{L_i\left(y^k_{t,i}\right)}, \quad \forall h_k, \quad k \in K, \quad t \in [t_0, T]\,. \tag{24}$$

$L_i\left(y^k_{t,i}\right)$ is estimated according to historical data. Note that the actual maintenance elapsed time of a type k check can be smaller or larger than $L_i\left(y^k_{t,i}\right)$. If additional slots are needed for an ongoing check, they will be created and updated according to (16).

Some airlines require a minimum number of days (d_k) between the start dates of two checks of the same type to prepare the maintenance resources, such as tools, workforce, aircraft spare parts and to avoid parallel peaks of workloads at the hangar, meaning that:

- If $d_k > 0$, there can be at most one aircraft starting a type k check at time t.
- If $d_k > 0$ and there is a type k check starting at t, no type k check is allowed to start in $[t, t + d_k - 1]$.

The requirement for the start date can be translated into the following equations for all time t:

$$\sum_{i=1}^{N_t} \chi^k_{t,i} \leq \begin{cases} 1 & \text{if } d_k > 0 \text{ and } \sum_{i=1}^{N_t} \chi^k_{\tau,i} = 0,\ \forall \tau \in [t - d_k, t-1] \\ M^k_t & \text{otherwise} \end{cases} \tag{25}$$

Eq. (25) indicates that there can be at most one type k check starting in $[t - d_k, t - 1]$ if $d_k > 0$, otherwise there can be at most M^k_t starting on a day.

3.4. Optimization Model

After the introduction of the objective function, state transition, and constraints, the optimization problem is to minimize (7), subject to constraints (13)–(25).

4. Methodology

In the S-AMCS optimization, the goal is to find a policy prescribing how to determine maintenance checks optimally in the face of uncertainty. However, it is, in general, difficult due to computational tractability. There are three main hindrances preventing us from computing such a policy:

H.1 Multi-dimensional state vector s_t, i.e., each aircraft has many attributes

H.2 Multi-dimensional action vector x_t, i.e., selecting different combinations of aircraft for letter checks

H.3 Very large outcome space, i.e., the number of outcome states is very large

In particular, **H.2** and **H.3** are closely correlated. For example, if the maintenance capacity of the day t is M_t^k for type k check, we would have the following number of possible actions for type k check:

$$\prod_{k \in K} \sum_{m_k=0}^{M_t^k} \frac{N_t!}{(N_t - m_k)!\ m_k!} \tag{26}$$

where $\frac{N_t!}{(N_t - m_k)!\ m_k!}$ represents the possible selections of aircraft for type k check. The number of outcome states on day t is the same as (26). As a result, the number of possible states on the day T is:

$$\prod_{t=t_0}^{T} \prod_{k \in K} \sum_{m_k=0}^{M_t^k} \frac{N_t!}{(N_t - m_k)!\ m_k!} \tag{27}$$

Even though for an example of two check types, A-check and C-check, a small fleet with ten aircraft, and one daily slot available for each check type, we would have 121 possible actions and associated outcome states on the first day, and more than 1.7 million possible sequences of actions just after three days.

A potential solution to address the S-AMCS problem formulated as MDP is dynamic programming (DP). Deng et al. (2020) addressed the deterministic AMCS (D-AMCS) optimization by defining maintenance check priority, applying a thrifty algorithm to estimate if the remaining slots will be sufficient, discretization, and state aggregation under the DP framework. However, an exact DP-based methodology is not suitable for S-AMCS since it relies on deterministic aircraft daily utilization and maintenance elapsed time. The DP-based methodology set forth by Deng et al. (2020) keeps a set of *workable* states for each day t using discretization and aggregation, from which it computes the *workable* states for $t + 1$. But in S-AMCS, once we make a maintenance decision on t, there will be only one state on $t + 1$ after revealing the new information. Working with a set of *workable* states and exploring the optimal sequence of actions is no longer possible.

In this section, we present a one-step lookahead approximate dynamic programming (ADP) methodology that uses simulations of aircraft utilization rates, maintenance elapsed times to estimate the future cost of each action via a thrifty policy, based on which we further determine high-quality current maintenance check decisions (for the day t). The lookahead ADP has a dynamic and adaptive nature and allows it to take advantage of the information that only becomes available between decision points.

This section presents the detail of the lookahead ADP methodology for S-AMCS. It begins with a brief introduction to the ADP concept in Subsection 4.1. Subsection 4.2 presents how Monte Carlo sampling is used to simulate uncertainty. Subsection 4.3 defines maintenance check priority for each aircraft and Subsection 4.4 defines basic rules for S-AMCS. After that, we describe two reference S-AMCS policies in Subsection 4.5 as benchmarks. Subsection 4.6 presents the detail of the lookahead ADP methodology. The last subsection (Subsection 4.7) shows an analysis of algorithm complexity.

4.1. Approximate Dynamic Programming

Approximate Dynamic Programming (ADP) is a modeling framework, based on an MDP model, that offers several strategies for tackling the curses of dimensionality in large, multi-period, stochastic optimization problems (Powell, 2011). ADP has been a research area of great interest for the last 30 years and is known under various names (e.g., reinforcement learning, neuro-dynamic programming). The idea is to make decisions by optimizing instant reward (*myopic* policy); or look ahead to future reward (*lookahead* policy) to make decisions; or use approximation techniques, such as simulation and machine learning to approximate the optimal policy (policy function approximation) or the value function (value function approximation), instead of solving (8). Policy function approximation (Novoa and Storer, 2009; McGrew et al., 2010) or value function approximation (Zhang and Adelman, 2009; Cai et al., 2009; Schmid, 2012; Medury and Madanat, 2013) usually requires a model, either parametric or non-parametric, to capture the features of a state.

In the S-AMCS, the fleet size and aircraft maintenance capacity vary during the planning horizon, e.g., new aircraft may phase-in, and old aircraft can phase-out/retire, and the maintenance slots are different on workdays than on public holidays. The *lookahead* ADP is so flexible that it works even though the fleet size or maintenance capacity changes over time. Besides, it does not necessarily require training of the model. The essence of the *lookahead* approach is a *lookahead* policy that can, ideally, find the optimal maintenance check action based on the estimations of the costs of all actions from a future period. The key is to approximate the value $V_{t+1}(s_{t+1})$ in (8) using Monte Carlo sampling and simulation.

4.2. Modeling of Uncertainty

Inspired by Rosenberger et al. (2000) and Gupta et al. (2003), we use stochastic simulation to capture uncertainty (generate information). A set of sample paths $\{w^n\}$, or so-called new information, is generated by Monte Carlo sampling (Vujic, 2018). Each sample path is a sequence of information $w^n = \{\omega^n_{t_0+1}, \omega^n_{t_0+2}, \dots, \omega^n_{T+1}\}$. We apply the classic Monte Carlo sampling on the sampling of aircraft daily FH and FC from historical data. For the aircraft daily FH, we first compute the mean (μ_i) and variance (σ_i) from historical aircraft daily utilization, then sample $\Delta\mathrm{fh}^{\omega}_{t,i}$ from normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, and $\Delta\mathrm{fc}^{\omega}_{t,i}$ also follows the same process.

On the other hand, we use Monte Carlo sampling for maintenance check elapsed time according to its empirical distribution. For example, given a set of C-check label and extra elapsed time (in working days) of aircraft i:

$$\textbf{C-1:}\quad \{-1,\ 0,\ 1,\ -2,\ 2,\ 0,\ 0,\ 0,\ 0,\ 2,\ -1,\ 0,\ 0,\ 0,\ 1,\ -1\} \tag{28}$$

where "-1" means C-1 finishes one day earlier, "-2" means C-1 ends two days earlier, "1" indicates that it takes one more day than expected, and "2" indicates that C-1 lasts two days longer than average. This gives the following empirical distribution:

Table 4: Empirical distribution of extra days for a check C-1.

Extra Days	**-2**	**-1**	**0**	**1**	**2**
Probability	0.0625	0.1875	0.5000	0.1250	0.1250

According to the distribution described in Table 4, we can employ Monte Carlo sampling for the extra days needed for C-1. Similarly, we do this for all the maintenance checks. After Monte Carlo sampling, the new information ω_{t+1} has the form of:

$$\omega_{t+1} = \left\{\omega^{\mathrm{A}}_{t+1}, \omega^{\mathrm{B}}_{t+1}, \omega^{\mathrm{C}}_{t+1}, \omega^{\mathrm{D}}_{t+1}\right\}, \tag{29}$$

$$\text{wherein} \quad \omega^{k}_{t+1} = \left\{\Delta L^{\omega}_{i}\left(y^{k}_{t,i}\right),\ \Delta \mathrm{fh}^{\omega}_{t+1,i},\ \Delta \mathrm{fc}^{\omega}_{t+1,i}\right\}, \quad t \in [t_0, T], \quad k \in K. \tag{30}$$

For each sample path $\{\omega_{t+1}, \omega_{t+2}, \ldots, \omega_{T+1}\}$, we make letter check decisions from t to T using pre-defined rules and policies (described in Subsection 4.3, 4.4 and 4.5.2), and we call this process *one simulation*.

4.3. Defining Maintenance Check Priority

As mentioned earlier, one major challenge in stochastic AMCS is the multi-dimensional action vector. According to (26), there are $\prod_{k\in K}\sum_{m_k=0}^{M^k_t}\frac{N_t!}{(N_t-m_k)!\ m_k!}$ actions on day t. To reduce the number of maintenance check actions, we propose a priority solution in the previous work (Deng et al., 2020), i.e., defining priorities for the fleet according to the rule of *earliest deadline first* for each check type. This rule does not specifically take any assumption on fleet size. It is common in maintenance scheduling and also convenient to implement in practice. Different from Deng et al. (2020), we use the term *expected remaining utilization* in S-AMCS to indicate the maintenance check deadline. The reason is that we can only estimate the *expected remaining utilization* according to the mean daily FH and FC of each aircraft and corresponding inspection interval. The *expected remaining utilization* unifies three different usage parameters of each aircraft (DY/FH/FC). It is defined by the fewest days to the next letter check:

$$R^{k}_{t,i} = \min\left\{R^{k\text{-DY}}_{t,i}, \quad R^{k\text{-FH}}_{t,i}, \quad R^{k\text{-FC}}_{t,i}\right\} \tag{31}$$

The $R^{k\text{-DY}}_{t,i}$, $R^{k\text{-FH}}_{t,i}$ and $R^{k\text{-FC}}_{t,i}$ refer to the *expected remaining utilization* with respect to each usage parameter and associated interval specified by the MPD:

$$R^{k\text{-DY}}_{t,i} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{r \leq I^{i}_{k\text{-DY}} - \mathrm{DY}^{k}_{t,i}\right\} \tag{32}$$

$$R^{k\text{-}\Psi}_{t,i} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{\sum_{\tau=t}^{t+r}\mathrm{fh}_{\tau,i} \leq I^{i}_{k\text{-}\Psi} - \Psi^{k}_{t,i}\right\} \tag{33}$$

where $\Psi \in \{\mathrm{FH},\ \mathrm{FC}\}$, $\psi \in \{\mathrm{fh},\ \mathrm{fc}\}$, $\psi_{\tau,i}$ and $\mathrm{fc}_{\tau,i}$ denote the average daily FH and FC of aircraft i; $\mathbb{N}$ is the set of natural numbers and k indicates the check type. After the *expected remaining utilization* is calculated, we sort $\left\{R_{t,i}^k\right\}_{i=1}^{N_t}$ in ascending order:

$$\tilde{R}_{t,1}^k,\ \tilde{R}_{t,2}^k,\ \tilde{R}_{t,3}^k,\ \ldots,\ \tilde{R}_{t,N_t}^k, \qquad \tilde{R}_{t,i}^k \leq \tilde{R}_{t,i+1}^k,\ \tilde{R}_{t,i}^k \in \left\{R_{t,i}^k\right\}_{i=1}^{N_t}. \tag{34}$$

The fleet is scheduled maintenance check according to the sequence presented in (34): aircraft with lower *expected remaining utilization* is given a higher check priority. Given hangar capacity M_t^k for the type k check on day t, after assigning priorities to the entire fleet, the number of actions for type k check of day t is reduced from $\mathrm{C}_{N_t}^0 + \mathrm{C}_{N_t}^1 + \cdots + \mathrm{C}_{N_t}^{M_t^k}$ to $M_t^k + 1$. The number of outcome states of each action of type k check is also reduced to 1. Since heavy maintenance (e.g., C-/D-check) is more restrictive and demanding in terms of resources, it has a higher priority than light maintenance (e.g., A-/B-check).

4.4. Basic Scheduling Rules for Stochastic Aircraft Maintenance Check Scheduling

We define the following basic rules for making maintenance check decisions before presenting the scheduling policies. These basic rules are the prerequisites for the stochastic AMCS:

(i) An aircraft i is considered to be allocated a type k check only if its *expected remaining utilization* is lower than the threshold R_{lb}^k (i.e., when $R_{t,i}^k \leq R_{\mathrm{lb}}^k$). This threshold is usually specified by airlines to prevent scheduling maintenance checks too often on the same aircraft.

(ii) If the number of type k check slots is sufficient, the aircraft with lowest *expected remaining utilization* $\tilde{R}_{t,1}^k = \min_i \left\{R_{t,i}^k\right\}$ has highest priority for type k check.

(iii) If aircraft i has a higher type k check priority than aircraft j $\left(R_{t,i}^k < R_{t,j}^k\right)$ but the remaining slots of type k check are only sufficient to accommodate a type k check for aircraft j rather than for aircraft i, swap the priorities between aircraft i and j for type k check.

(iv) If an aircraft reaches its maximum utilization of type k check on the day t and there is no available slot, additional slots will be created until the type k check is finished.

4.5. Reference Scheduling Policies

To address the S-AMCS, we propose to use ADP to schedule aircraft maintenance checks based on fleet status, following pre-defined policies. In this subsection, we introduce two simple scheduling policies, the *myopic* policy and *thrifty* policy. These two policies are simple scheduling policies that work as benchmarks for our lookahead policy. The *myopic* policy is a greedy approach that serves as an upper bound for the average aircraft utilization and a lower bound for the total number of maintenance checks. On the contrary, the *thrifty* policy is a conservative approach that provides a lower bound for the average aircraft utilization and an upper bound for the total number of maintenance checks.

4.5.1. Myopic Policy

The *Myopic* policy is one of the most elementary policies. It makes a maintenance check decision according to the minimum immediate contribution, without looking into the future cost. For each day t, the *myopic* policy enables us to make maintenance check decisions only if an aircraft reaches the inspection interval of type k check. This is equivalent to assuming $V_{t+1}(s_{t+1}) = 0$ in (8):

$$x_t^* = \underset{x_t \in X_t}{\operatorname{argmin}} \left\{ C_t\left(s_t, x_t\right) \right\} \tag{35}$$

where X_t denotes the set of actions associated with s_t, $X_t = \{\mathscr{X}^\pi\left(s_t\right)\}$. The *myopic* policy runs very fast and if it results in no additional slot in S-AMCS (e.g., there is infinite aircraft maintenance capacity), then (35) is already the optimal policy. However, considering the limited maintenance capacity in practice, *myopic* policy often leads to poor solutions in terms of creating lots of additional maintenance slots.

4.5.2. Thrifty Policy

The *thrifty* policy is a conservative policy that schedules maintenance checks whenever there is an available slot (Deng et al., 2020) according to the maintenance check priority of all aircraft and all check types. If several hangars fit the most maintenance checks, we choose the hangars with the closest value. Similar to the *myopic* policy, the *thrifty* policy makes maintenance check decisions without looking into the future cost. It only checks whether the available slots from t matches the mean maintenance check elapsed time (the actual elapsed time is only known at $t+1$ after a maintenance check is decided). It runs even faster than the *myopic* policy but results in low aircraft utilization and a relatively large number of maintenance checks.

4.6. Lookahead Approximate Dynamic Programming

The lookahead approximate dynamic programming (ADP) methodology consists of two parts, a dynamic programming framework and a hybrid lookahead policy. The dynamic programming framework is the same as described in Deng et al. (2020), consisting of a forward induction approach, a priority solution, and the basic scheduling rules mentioned in Subsection 4.4 for AMCS. The hybrid lookahead policy combines deterministic and stochastic forecasts.

To address the S-AMCS, we need to solve the following equation:

$$x_t^* = \underset{x_t \in X_t}{\operatorname{argmin}} \left\{ C_t\left(s_t, x_t\right) + \gamma \overline{V}_t\left(s_t\right) \right\} \tag{36}$$

It means that we use a *lookhead policy* to generate $\overline{V}_t\left(s_t\right)$ as an approximation of $V_t\left(s_t\right)$ in (8) based on simulations of future aircraft utilization and maintenance check elapsed time and then make decision by solving (36). Since there are limited maintenance resources and capacities in the stochastic AMCS, creating extra maintenance slots beyond the maintenance capacity of airlines is one of the major operating costs. Therefore, we first use the *thrifty* policy discussed in Deng et al. (2020) to explore the future and estimate the number of additional maintenance slots that would be needed if an action is taken:

$$g_k\left(\hat{s}_t, t + t_h\right) = \sum_{i=1}^{N_t} \sum_{\tau=t}^{t+t_h} \hat{\eta}_{\tau,i}^k, \quad k \in K. \tag{37}$$

where $\hat{\eta}^k_{\tau,i}$ denotes the number of additional slots created on day τ, without knowing any information from $t+1$, and t_h is a positive integer. Note that computing $g_k(\hat{s}_t, t+t_h)$ in (37) requires $\hat{s}_t$ ($\hat{s}_t = \mathcal{S}^X(s_t, x_t)$), the mean aircraft daily utilization, and the mean elapsed time for the entire fleet. Obtaining $g_k(\hat{s}_t, t+t_h)$ is equivalent to applying the *thrifty* policy from $t+1$ to $t+t_h$.

As C-checks happen every 18–24 months and D-checks occur every 5–6 years, and the daily utilization of an aircraft follows a normal distribution, in the long term, the sequence of daily utilization can be considered independent and identically distributed. According to the law of large numbers, the observed cumulative utilization of an aircraft since its previous C-/D-check is very close to the mean daily utilization multiplying the elapsed days. Hence, we can use the average daily utilization of each aircraft to simulate when the coming C-/D-checks take place. Besides, C-/D-checks are usually not allowed to perform during the commercial peak season in practice, such as summer, Easter, or Christmas holidays, indicating that C-/D-checks are jammed in the non-commercial periods. Therefore, similar to the aircraft daily utilization, the impact of uncertainty from C-/D-check elapsed time can be significantly diminished. However, (37) cannot predict the future extra maintenance slots for the other check types that happen more often, e.g., A-/B-checks. The future period $[t, t+t_h]$ to look ahead is too large in (37), and A-/B-check occurs too often to anticipate using only the mean aircraft daily utilization. Hence, to provide a more accurate prediction on the extra maintenance slots for A-/B-check, we propose a hybrid policy combining deterministic and stochastic forecasts:

- *Step 1*: Determine the one-step optimal C- and D-check actions based on deterministic forecasts
- *Step 2*: Determine the one-step optimal A- and B-check actions using stochastic forecasts

4.6.1. Determine the One-Step Optimal C- and D-Check Actions using Deterministic Forecasts

Before determining the optimal C- and D-check actions, it is worth mentioning that wasting an available maintenance slot at present can result in a shortage of maintenance slots in the future. From the perspective of an airline, skipping a maintenance slot on the day t_1 means that some technicians are idle (not performing maintenance works), and the airline still needs to pay for those technicians. On the other hand, when we create one extra slot on the day t_2 ($t_2 > t_1$), the airline has to spend more money to compensate for the extra work from the technicians or to subcontract the maintenance check. Therefore, we want to penalize both the waste of an available slot on day t and the extra costs for creating more slots in $[t+1, t+t_h]$. We give a penalty to the objective values when all the following conditions are met:

C.1 There are sufficient slots for a type k check, namely, $\exists i,\ R^k_{t,i} \leq R^k_{\text{lb}}$ and constraint (24) holds.

C.2 $g_k(\hat{s}_t, t+t_h) > 0$, i.e., there is at least one extra maintenance slot of type k check created in $[t, t+t_h]$.

C.3 There is no action of type k check, i.e., $\sum_{i=1}^{N_t} \chi^k_{t,i} = 0$.

According to this logic, we use the following approximation for $V_t(s_t)$ in (8):

$$V_t(s_t) \approx \overline{V}_t^{(1)}(s_t) = \sum_k \Bigg(\lambda g_k(\hat{s}_t, t+t_h) + \underbrace{\max_{R_{t,i}^k \le R_{\text{lb}}^k} \left\{ \text{sgn}\left(\frac{\sum_{\tau=t}^{t+L_i(y_{t,i}^k)} M_{\tau,h_k}^k}{L_i(y_{t,i}^k)} - \chi_{t,i}^k \right) \right\}}_{\textbf{C.1}} \underbrace{\text{sgn}(g_k(\hat{s}_t, t+t_h))}_{\textbf{C.2}} \underbrace{\left[1 - \text{sgn}\left(\sum_{i=1}^{N_t} \chi_{t,i}^k \right) \right]}_{\textbf{C.3}} \xi \Bigg) \quad (38)$$

where $\overline{V}_t^{(1)}(s_t)$ corresponds to the *deterministic forecast*, i.e., a forecast that does not include expression of the associated uncertainty, following Powell (2011), where λ is a large constant (cost per extra slot) to prevent creating unnecessary additional maintenance slots, ξ is a large constant to prevent the waste of an available slot, and "sgn" is the sign function:

$$\text{sgn}(\alpha) = \begin{cases} -1 & \text{if } \alpha < 0 \\ 0 & \text{if } \alpha = 0 \\ 1 & \text{if } \alpha > 0 \end{cases} \quad (39)$$

We only keep the optimal C- and D-check actions for day t:

$$x_{t,\text{det}}^* = \left\{ x_{t,\text{det}}^{\text{A}*}, x_{t,\text{det}}^{\text{B}*}, x_{t,\text{det}}^{\text{C}*}, x_{t,\text{det}}^{\text{D}*} \right\} = \underset{x_t \in X_t}{\text{argmin}} \left\{ C_t(s_t, x_t) + \gamma \overline{V}_t^{(1)}(s_t) \right\} \quad (40)$$

$$x_t^{\text{C}*} = x_{t,\text{det}}^{\text{C}*}, \quad x_t^{\text{D}*} = x_{t,\text{det}}^{\text{D}*}. \quad (41)$$

In (40), "det" is the abbreviation for "deterministic". For C-/D-check, we use the deterministic forecasts, namely, the mean daily utilization and maintenance elapsed time, to assess whether the maintenance slots are sufficient in the future in the *thrifty* algorithm for $[t+1, t+t_h]$, then determine the best C- and D-check action. In this way, we tremendously reduce ADP algorithm complexity for prediction of coming C-/D-checks. After obtaining the optimal C-/D-check actions from (40) and (41), we fix $x_t^{\text{C}*}$ and $x_t^{\text{D}*}$.

4.6.2. Determine the One-Step Optimal A- and B-Check Actions using Stochastic Forecasts

Since the aircraft A-/B-check occurs once every few months, the uncertainty in daily aircraft utilization can significantly impact the dates of A-/B-checks. We can rely on the stochastic forecasts to estimate when the A- and B-checks are likely to occur in a shorter future period $[t+1, t+t_l]$ ($t_l \ll t_h$). For each maintenance check action, we carry out Monte Carlo simulations:

$$w_{t+1}^n = \left\{ \omega_{t+1}^n,\ \omega_{t+2}^n,\ \ldots, \omega_{t+t_l+1}^n \right\}, \quad n = 1, 2, \ldots, n_{\text{sample}},\ t \in [t_0, T]. \quad (42)$$

$$g_k^\omega(\hat{s}_t, t+t_l, w_{t+1}^n) = \sum_{i=1}^{N_t} \sum_{\tau=t}^{t+t_l} \eta_{\tau,i}^k(\omega_{\tau+1}^n), \quad k \in K. \quad (43)$$

$$G_k(\hat{s}_t, t+t_l) = \frac{1}{n_{\text{sample}}} \sum_{n=1}^{n_{\text{sample}}} g_k^\omega(\hat{s}_t, t+t_l, w_{t+1}^n) \quad (44)$$

Similar to (38), we use the following approximation for $V_t(s_t)$ in (8):

$$V_t(s_t) \approx \overline{V}_t^{(2)}(s_t) = \sum_k \Bigg(\lambda G_k(\hat{s}_t, t+t_l) + \max_{R^k_{t,i} \le R^k_{\text{lb}}} \left\{ \text{sgn}\left(\frac{\sum_{\tau=t}^{t+L_i\left(y^k_{t,i}\right)} M^k_{\tau,h_k}}{L_i\left(y^k_{t,i}\right)} - \chi^k_{t,i} \right) \right\} \text{sgn}\left(G_k(\hat{s}_t, t+t_l)\right) \left[1 - \text{sgn}\left(\sum_{i=1}^{N_t} \chi^k_{t,i}\right)\right] \xi \Bigg) \tag{45}$$

In contrast to $\overline{V}_t^{(1)}(s_t)$, $\overline{V}_t^{(2)}(s_t)$ uses the sample realizations to approximate $V_t(s_t)$. $\overline{V}_t^{(2)}(s_t)$ corresponds to the *stochastic forecast* (Powell, 2011). After that, we determine the optimal A- and B-check actions:

$$x_t^* = \left\{x_t^{\text{A}*}, x_t^{\text{B}*}, x_t^{\text{C}*}, x_t^{\text{D}*}\right\} = \underset{\left\{x_t^{\text{A}}, x_t^{\text{B}}, x_t^{\text{C}*}, x_{t,}^{\text{D}*}\right\} \in X_t}{\text{argmin}} \left\{ C_t\left(s_t, \left\{x_t^{\text{A}}, x_t^{\text{B}}, x_t^{\text{C}*}, x_{t,}^{\text{D}*}\right\}\right) + \gamma\lambda \overline{V}_t^{(2)}(s_t) \right\} \tag{46}$$

Note that we use the deterministic forecasts from $[t+1, t+t_h]$, and stochastic forecasts from $[t+1, t+t_l]$ to make the maintenance check decision only for the day t. After that, we move to $t+1$ and update the state according to new information. We repeat the same process on $t+1$ to determine the maintenance check action for the day $t+1$. We call (40)–(46) a lookahead ADP methodology. The detail of the lookahead ADP methodology is presented in Algorithm 1.

4.7. Algorithm Complexity

To assess the algorithm complexity of the lookahead ADP methodology, we count how many times the state transition function (12) is called to find the best action x_t^* of the day t. For an action x_t, we mean a set of maintenance check actions for all check types of the day t. For comparison purpose, we also provide the algorithm complexity analysis for the *myopic* and *thrifty* policies.

There is only one state s_t on a day t in S-AMCS, and s_t has at most n_{act} actions after sorting the priorities for all maintenance check types and all aircraft:

$$n_{\text{act}} = \prod_{k \in K} n^k_{\text{act}} = \prod_{k \in K} \left(M_t^k + 1\right) \tag{47}$$

In the *myopic* policy, we have to check all actions and find the one resulting the minimum daily contribution, without looking into the future cost. It means that if there are n_{act} actions on the day t, we have to call (12) n_{act} times in any case in the *myopic* policy. Hence, the algorithm complexity of the *myopic* policy is n_{act}.

For *thrifty* policy, we allocate the maintenance checks whenever there are sufficient available maintenance slots based on the mean elapsed time of the maintenance checks. Namely, we check the hangar capacity first and see how many checks the hangars can accommodate. We then choose the action that fits the most maintenance checks in the hangars following the priorities defined in Subsection 4.3. If several hangars fit the most maintenance checks, we choose the hangars with the closest value. Therefore, we just need to call (12) only once on the day t in the *thrifty* policy and the algorithm complexity of *thrifty* policy is 1.

In the lookahead ADP methodology, it makes the aircraft maintenance check decisions in two steps. It first determines the one-step optimal actions for aircraft C- and D-checks, and then for aircraft A- and B-checks.

Algorithm 1 A Lookahead ADP Methodology for Stochastic Aircraft Maintenance Check Scheduling Optimization

Initialize $\hat{s}_{t_0-1}$ ▷ Initial input data

$t \leftarrow t_0$

$\hat{s}_{t-1} \leftarrow \hat{s}_{t_0-1}$

procedure APPROXIMATE DYNAMIC PROGRAMMING

  while $t_0 < T$ **do**

    $\omega_t \longleftarrow \left\{\Delta L_i^{\omega}\left(y_{t-1,i}^{k}\right),\ \Delta \mathrm{fh}_{t,i}^{\omega},\ \Delta \mathrm{fc}_{t,i}^{\omega}\right\}_{i=1}^{N_t}$ ▷ Arrival of new information

    $s_t \leftarrow \mathcal{S}^W\left(\hat{s}_{t-1}, \omega_t\right)$ ▷ Pre-Decision update

    procedure FIND THE ONE-STEP OPTIMAL MAINTENANCE CHECK ACTION

      $X_t \leftarrow \{x_t | x_t = \mathscr{X}^{\pi}\left(s_t\right)\}$ ▷ Generate a set of actions according to Eq. (3)

      Compute and sort aircraft remaining utilization using Eq. (31)–(34) ▷ Define maintenance check priority

      procedure DETERMINE THE BEST C- AND D-CHECK DECISIONS

        $g_k\left(\hat{s}_t, t+T\right) \leftarrow \sum_{i=1}^{N_t}\sum_{\tau=t}^{t+T}\hat{\eta}_{\tau,i}^{k}, \quad k \in K$

        $\overline{V}_t^{(1)}\left(s_t\right) \leftarrow$ Eq. (38)

        $\left\{x_{t,\mathrm{det}}^{\mathrm{A}*}, x_{t,\mathrm{det}}^{\mathrm{B}*}, x_{t,\mathrm{det}}^{\mathrm{C}*}, x_{t,\mathrm{det}}^{\mathrm{D}*}\right\} \leftarrow \mathrm{argmin}_{x_t \in X_t}\left\{C_t\left(s_t, x_t\right) + \gamma \overline{V}_t^{(1)}\left(s_t\right)\right\}$ ▷ $\hat{s}_t = \mathcal{S}^X\left(s_t, x_t\right)$

        $x_t^{\mathrm{C}*} \leftarrow x_{t,\mathrm{det}}^{\mathrm{C}*}, \quad x_t^{\mathrm{D}*} \leftarrow x_{t,\mathrm{det}}^{\mathrm{D}*}$ ▷ Find the optimal C- and D-check actions

      end procedure

      procedure DETERMINE THE BEST A- AND B-CHECK DECISIONS

        $w_{t+1}^{n} = \left\{\omega_{t+1}^{n},\ \omega_{t+2}^{n},\ \ldots, \omega_{t+t_l+1}^{n}\right\} \quad n = 1, 2, \ldots, n_{\mathrm{sample}},\ t \in [t_0, T]$ ▷ Monte Carlo sampling

        $g_k^{\omega}\left(\hat{s}_t, t+t_l, w_{t+1}^{n}\right) \leftarrow \sum_{i=1}^{N_t}\sum_{\tau=t}^{t+t_l}\eta_{\tau,i}^{k}\left(\omega_{\tau+1}^{n}\right)$ ▷ Simulation

        $G_k\left(\hat{s}_t, t+t_l\right) \leftarrow \frac{1}{n_{\mathrm{sample}}}\sum_{n=1}^{n_{\mathrm{sample}}} g_k^{\omega}\left(\hat{s}_t, t+t_l, w_{t+1}^{n}\right)$

        $\overline{V}_t^{(2)}\left(s_t\right) \leftarrow$ Eq. (45)

        $\left\{x_t^{\mathrm{A}*}, x_t^{\mathrm{B}*}, x_t^{\mathrm{C}*}, x_t^{\mathrm{D}*}\right\} \leftarrow \mathrm{argmin}_{\left\{x_t^{\mathrm{A}}, x_t^{\mathrm{B}}, x_t^{\mathrm{C}*}, x_{t,}^{\mathrm{D}*}\right\} \in X_t}\left\{C_t\left(s_t, \left\{x_t^{\mathrm{A}}, x_t^{\mathrm{B}}, x_t^{\mathrm{C}*}, x_{t,}^{\mathrm{D}*}\right\}\right) + \gamma\lambda\overline{V}_t^{(2)}\left(s_t\right)\right\}$

      end procedure

      $x_t^{*} \leftarrow \left\{x_t^{\mathrm{A}*}, x_t^{\mathrm{B}*}, x_t^{\mathrm{C}*}, x_t^{\mathrm{D}*}\right\}$ ▷ Optimal maintenance check action found

      $\hat{s}_t^{*} \leftarrow \mathcal{S}^X\left(s_t, x_t^{*}\right)$ ▷ Post-Decision update

      $\hat{s}_t \leftarrow \hat{s}_t^{*}$

    end procedure

    $t \leftarrow t + 1$

  end while

end procedure

In the first step, we apply the *thrifty* algorithm to compute the number of extra maintenance slots for the period of $[t+1, t+t_h]$. Since we only need to call (12) once for each day in the *thrifty* algorithm, computing the number of extra maintenance slots for $[t+1, t+t_h]$ is equivalent to calling (12) t_h times. Multiplying t_h with the number of actions n_{act} implies the algorithm complexity of the first step:

$$n_{\text{act}} t_h \tag{48}$$

In the second step of the lookahead ADP methodology, we fix the optimal C- and D-check actions obtained from the previous step, then use Monte Carlo simulations to estimate the number of extra A- and B-check slots for the future period $[t+1, t+t_l]$. For each sample path, we use the *thrifty* algorithm to compute the extra slots, that is, running the *thrifty* algorithm on $[t+1, t+t_l]$. It means that we call (12) t_l times for each sample path. The total number of sample paths n_{sample} makes us call (12) $n_{\text{sample}} t_l$ times for each action. Since we already determine the optimal aircraft C- and D-check decisions in the first step, the number of actions in the second step, $n_{\text{act}}^{\text{A}} \times n_{\text{act}}^{\text{B}}$, is smaller than n_{act}. The algorithm complexity of the second step is:

$$n_{\text{act}}^{\text{A}} n_{\text{act}}^{\text{B}} n_{\text{sample}} t_l \tag{49}$$

Summing (48) and (49) gives the following algorithm complexity of determining the optimal action on a day t in the lookahead ADP methodology:

$$n_{\text{act}} t_h + n_{\text{act}}^{\text{A}} n_{\text{act}}^{\text{B}} n_{\text{sample}} t_l < n_{\text{act}} \left(t_h + n_{\text{sample}} t_l\right) \tag{50}$$

We can see that the lookahead ADP methodology has polynomial time complexity, which is suitable for practical implementation to the S-AMCS problem.

5. Case Study

The proposed ADP methodology for S-AMCS is evaluated using the aircraft maintenance data and daily utilization from a European airline (Deng, 2020). The test fleet is the Airbus A320 family (A319, A320, A321-1, and A321-2), consisting of 40-50 aircraft. The airline distributes the tasks within B-check into successive A-checks (no B-check), merges the D-checks in C-checks, and labels them as heavy C-checks. Table 5 presents the associated inspection interval of each aircraft type. Two case studies are presented: the first case focuses on September 25$^{\text{th}}$ 2017–December 31$^{\text{st}}$ 2020 and has aircraft type A319, A320, and A321-1 since we have the C-check schedule of this period from the airline for comparison; the second case focuses on the period of March 20$^{\text{th}}$ 2019 to December 31$^{\text{st}}$ 2022 and has all four aircraft types. For each test case, there are five policies/methodologies tested:

M.1 Lookahead ADP with deterministic and stochastic forecasts, labeled as "ADP-DS"

M.2 The optimal deterministic AMCS schedule planned by Deng et al. (2020), labeled as "DP-based"

M.3 Myopic policy, labeled as "Myopic"

M.4 Thrifty policy, labeled as "Thrifty"

M.5 Lookahead ADP methodology itself using only deterministic forecasts, labeled as "ADP-D"

The ADP-D includes only (38)—(40) and make the optimal AMCS decision $x_t^* = x_{t,\text{det}}^*$. We benchmark the outcomes from **M.1** against the results from **M.2**—**M.5**.

Table 5: Maintenance check intervals of Airbus A319, A320 and A321 (AIRBUS, 2017).

Aircraft	**A-Check**			**C-Check**			**D-Check**
Type	**DY**	**FH**	**FC**	**DY**	**FH**	**FC**	**DY**
A319	120	750	750	730	7500	5000	2192
A320	120	750	750	730	7500	5000	2192
A321-1	120	750	750	730	7500	5000	2192
A321-2	120	750	750	1096	12000	8000	2192

5.1. Maintenance Actions

The airline has at most two A-check slots per workday ($\max\left\{M_t^{\text{A}}\right\} = 2$) and three C-check slots per day during the C-check period ($\max\left\{M_t^{\text{C}}\right\} = 3$), but there are at least three days between the start dates of two successive C-checks ($d_{\text{C}} = 3$). The airline needs these three days to prepare the maintenance tools. It means that there could be at most one C-check starting on a day. According to the requirements of our airline partner, D-checks are merged within C-check in the following pattern:

$$\text{C-1}, \quad \text{C-2}, \underbrace{\text{C-3}}_{\text{D-check}}, \quad \text{C-4}, \quad \text{C-5}, \underbrace{\text{C-6}}_{\text{D-check}}, \text{C-7}, \quad \text{C-8}, \underbrace{\text{C-9}}_{\text{D-check}}, \ldots \tag{51}$$

Moreover, D-check has to be performed within the interval of 2192 DY. The maximum of two A-checks slots on weekdays and the possibility of merging A- into C-check together lead to 12 possible combinations of total daily A- and C-check actions, as shown in Table 6.

5.2. Key Performance Indicators

To discuss the results, we use a set of key performance indicators (KPIs) for each type of letter check. These KPIs are the average FH of the entire fleet, the total number of maintenance checks, the total number of extra slots, and the average computation time of making the optimal decision for a day.

Table 6: Possible aircraft maintenance check actions on a day t.

Maintenance Check Action	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Number of A-Checks	0	0	1	1	2	2	3	3	4	4	5	5
Number of C-Checks	0	1	0	1	0	1	0	1	0	1	0	1

To validate the proposed lookahead ADP methodology, we use 100 test runs for each test case. Each test run corresponds to one test sample path generated using Monte-Carlo sampling, from which we can see how well the lookahead ADP copes with uncertainty and how robust this methodology is. After one test run, we obtain a set of associated average FH of the fleet, the total number of maintenance checks, the total number of extra slots, and the average computation time of making the optimal decision for a day. Each of the KPIs is the mean of 100 test runs. For example, the KPI average FH of the entire fleet is the mean of 100 average FH resulting from 100 test runs. And this also applies to the calculation of other KPIs for all the policies/methodologies to be tested.

To simulate the performance of the DP-based methodology over the test sample paths, we first plan the optimal maintenance check schedule for the deterministic AMCS model and then test the optimal schedule over the sample paths and adjust the A-/C-check when necessary. An additional maintenance slot is created every time the maintenance schedule becomes unfeasible.

For the other policies/methodologies, we plan the optimal maintenance check day by day, from the first day to the last day of the planning horizon, considering the new information provided per day, according to the sample path. The test cases are further used to support a sensitivity analysis on some of the model parameters. All the aircraft A- and C-check schedules are generated using the same input data and under the same operational constraints of the airline, as described in Deng et al. (2020).

5.3. Model Parameters

We assign 21 and 210 to $R^{\mathrm{A}}_{\mathrm{lb}}$ and $R^{\mathrm{C}}_{\mathrm{lb}}$ following the current practice of our airline partner. λ is given 10^5 suggested by our airline partner based on the results presented in Deng et al. (2020). Setting $\lambda = 10^5$ can avoid creating unnecessary additional maintenance slots. We assign 10^{20} to ξ to penalize the action of wasting available maintenance slots of a day when the lookahead policy predicts a non-zero extra maintenance slot in the future. The reason for having $\xi \gg \lambda$ is that, in the situation of wasting an available slot of a day t_1 when the lookahead policy predicts an extra maintenance slot on a day $t_2 > t_1$, the airline still has to pay for technicians for being idle on t_1 and spend a higher cost to compensate the extra work from technicians on t_2. Therefore, we use $\xi = 10^{20}$ to prevent this circumstance. For ADP-DS, we use 50 sample paths in Monte Carlo simulation to evaluate each action, i.e., $n_{\text{sample}} = 50$ (600 in total for 12 actions). For ADP-D, we use only the mean daily aircraft utilization and the mean maintenance check elapsed time.

Both test cases are conducted using parallel computing on a quad-core workstation. We look six months ahead for A-check ($t_l = 183$), and four years ahead for C-check ($t_h = 1461$) to estimate the cost of creating additional maintenance slots. The reason is that if the algorithm allocates an A-/C-check to an aircraft, we can always anticipate the next check. A summary of model parameters is presented in Table 7.

5.4. Outcomes for the Test Case 2017–2020

We first look at the KPIs of the test case 2017–2020. An ideal schedule/S-AMCS policy should result in better KPIs, i.e., higher average FH, fewer total checks, and fewer extra maintenance slots for both check

Table 7: Model parameters for Stochastic AMCS optimization

Parameters	**Description**	**Value**	**Unit**
R^{A}_{lb}	A utilization threshold to prevent scheduling A-check too often (Deng et al., 2020)	21	day
R^{C}_{lb}	A utilization threshold to prevent scheduling C-check too often (Deng et al., 2020)	210	day
γ	Discount factor for Stochastic AMCS model	1	—
λ	Cost of creating an additional maintenance slot	10^5	FH
ξ	Penalty for the waste of an available maintenance slot	10^{20}	FH
n_{sample}	The number of sample paths for Monte Carlo simulations	50	—
t_l	A future time period for A-check to look ahead in rolling horizon	183	day
t_h	A future time period for C-check to look ahead in rolling horizon	1461	day

types than the maintenance check schedule of the airline. As shown in Table 8, the schedules from DP-based methodology and the *myopic* policy both result in more than 90 extra C-checks slots and 20 extra A-checks slots on average for the 100 test sample paths, compared with the C-check schedule and A-check estimation of the airline (15 additional slots for each check type). It means that the optimal A- and C-check schedule from the deterministic AMCS generated by the DP-based methodology is not robust to uncertainty. Without looking into the future cost, the *myopic* policy is too greedy in A- and C-check scheduling. Although these two approaches achieve higher aircraft utilization for both check types, the airline has to face extra costs to create additional maintenance capacity if any of the two approaches is executed.

Conversely, the *thrifty* policy does not need to create any extra maintenance slot for all 100 test sample paths. The *thrifty* policy is too conservative, and the associated mean average FH for C-check is 6.7% lower than the C-check schedule of the airline. For A-check, the associated mean average FH is 17.5% lower. There is a trade-off between aircraft utilization and the number of extra slots. The *thrifty* policy is more robust to uncertainty, yet at the cost of achieving a lower aircraft utilization.

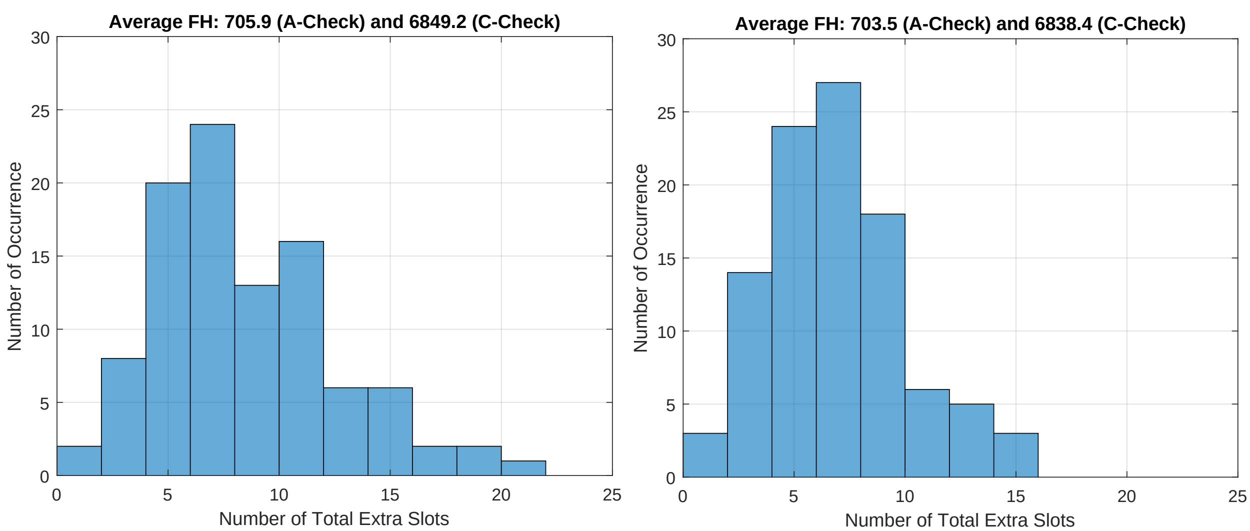
The lookahead ADP methodology with only deterministic forecasts, ADP-D, leads to higher mean average aircraft utilization and fewer extra maintenance slots for both check types and 100 test sample paths, compared with the C-check schedule and A-check estimation of the airline. It outperforms the optimal schedule generated by the DP-based methodology and the *myopic* and *thrifty* policies.

The proposed lookahead ADP methodology that combines deterministic and stochastic forecasts, ADP-DS, creates the second least mean extra slots (after the *myopic*), 0.8 extra slots on average for A-check, and 5.7 for C-check. The associated mean average FH for A-check/C-check is 8.5 and 191.6 higher, respectively, compared with the C-check schedule and A-check estimation of the airline. Besides, the differences in mean average FH between ADP-D and ADP-DS is only 0.34%/0.16% for A-/C-check, meaning that these two approaches are equivalently promising in terms of aircraft utilization. Even so, due to the stochastic forecasts on extra A-check slots, the ADP-DS leads to 50% fewer A-checks and 12.3% fewer C-checks than the ADP-D.

Figure 3 shows the distributions of total extra slots under the ADP-D and ADP-DS for the 100 test

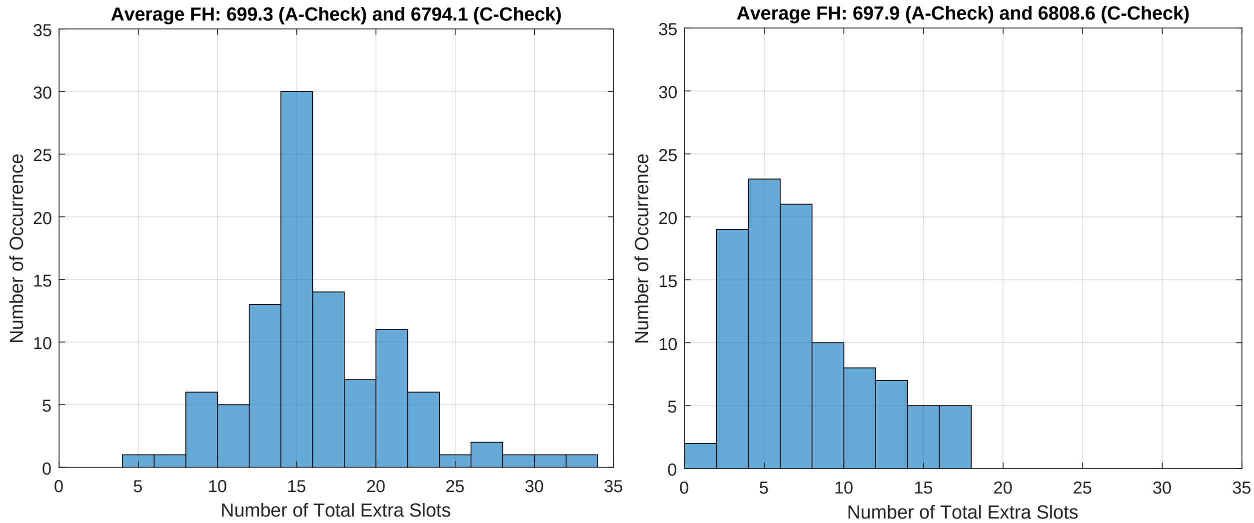
Table 8: Comparison of KPIs for September 25th 2017–December 31st 2020 for 100 test sample paths. The numbers labeled with "*" are estimated or extrapolated according to the historical maintenance data of the airline. ADP-D represents the lookahead ADP with only deterministic forecasts. ADP-DS represents the lookahead ADP with both deterministic and stochastic forecasts.

KPI 2017–2020 (1194 days)		**Airline Schedule**	**Stochastic Results (100 test runs)**				
			DP-based	**Myopic**	**Thrifty**	**ADP-D**	**ADP-DS**
C-check	Mean Average FH	6646.8	6785.4	7142.1	6200.6	6849.2	6838.4
	Mean Extra Slots	15	90.4	368.4	0.0	6.5	5.7
	Mean Total Checks	88	77.0	75.3	83.1	79.2	79.4
A-check	Mean Average FH	695.0*	713.3	744.6	573.6	705.9	703.5
	Mean Extra Slots	$\geq$ 15*	20.4	367.3	0.0	1.6	0.8
	Mean Total Checks	750*	727.0	698.4	893.6	733.6	735.9
Mean Total Extra Slots		30*	110.8	735.7	0.0	8.1	6.5
95% Confidence Interval		—	[109.41, 112.19]	[732.23, 739.17]	[0, 0]	[7.32, 8.88]	[5.88, 7.12]
Computation Time/day [s]		—	0.02	0.09	0.05	0.35	2.63



(a) (b)

Figure 3: Distributions of total extra slots under two methodologies for the test case 2017–2020, under 100 test sample paths: (a) Distribution of total extra slots under the ADP-D (lookahead ADP using only deterministic forecasts); (b) Distribution of total extra slots under the ADP-DS (lookahead ADP using both deterministic and stochastic forecasts).



(a) (b)

Figure 4: Distributions of total extra slots under two methodologies for the test case 2019–2022, under 100 test sample paths: (a) Distribution of total extra slots under the ADP-D (lookahead ADP using only deterministic forecasts); (b) Distribution of total extra slots under the ADP-DS (lookahead ADP using both deterministic and stochastic forecasts).

runs. We can observe that ADP-DS creates no more than 15 additional slots for all the test runs, and in 86% of test runs, it uses less than ten extra slots. For ADP-D, the airline may need to create more than 20 additional slots to cope with the uncertainty, and the chance of creating more than ten extra slots is higher than 33%. Therefore, according to the results of 100 test sample paths, ADP-DS outperforms ADP-D in fewer additional slots for both check types and almost the same average aircraft utilization. Furthermore, Table 9 shows that a Student's t-test rejects the null hypothesis that the two methods have similar performance, at a 5% significance level. That is, the outcomes from the two methods do have mean values that significantly differ from each other.

Table 9: Student's t-test on the results from ADP-D and ADP-DS for the test case 2017–2020.

t-value	**p-value**	**Degrees of Freedom**	**Pooled Estimate of the Population Standard Deviation**
3.0477	0.0030	99	5.0859

5.5. Outcomes for the Test Case 2019–2022

As mentioned in the previous test case, an ideal schedule/S-AMCS policy should results in better KPIs, i.e., higher average FH and fewer total checks for both check types than the maintenance check schedule of the airline while creating fewer extra maintenance slots. Table 10 shows that the KPIs of the second test case follow a similar trend to the first test case. The *myopic* policy results in the highest aircraft utilization, yet creating the most extra slots for both check types. The *thrifty* policy leads to the lowest aircraft utilization

Table 10: Comparison of KPIs for March 20th 2019–December 31st 2022 for 100 test sample paths. The numbers labeled with "*" are estimated or extrapolated according to the historical maintenance data of the airline. ADP-D represents the lookahead ADP with only deterministic forecasts. ADP-DS represents the lookahead ADP with both deterministic and stochastic forecasts.

KPI 2019–2022		**Airline**	**Stochastic Results (100 test runs)**				
(1383 days)		**Estimation**	**DP-based**	**Myopic**	**Thrifty**	**ADP-D**	**ADP-DS**
C-check	Mean Average FH	6700.0*	6920.9	7469.4	6361.7	6794.1	6808.6
	Mean Extra Slots	$\geq$ 20.0*	20.9	426.8	0.0	4.0	4.0
	Mean Total Checks	100*	90.0	88	94.0	90.8	90.6
A-check	Mean Average FH	695.0*	708.8	744.2	614.1	699.3	697.9
	Mean Extra Slots	$\geq$ 20.0*	19.4	517.5	0.9	12.0	3.0
	Mean Total Checks	1030*	1003.0	959.6	1151.8	1017.9	1019.9
Mean Total Extra Slots		$\geq$ 40.0*	40.3	944.3	0.9	16.0	7.0
95% Confidence Interval		——	[38.98, 41.62]	[940.15, 948.51]	[0.72, 1.08]	[15.02, 16.98]	[6.19, 7.81]
Computation Time/day [s]		——	0.02	0.09	0.05	0.35	2.63

and the least extra slots as expected. In the second test case, the optimal schedule from deterministic AMCS obtained from the DP-based methods becomes more robust to uncertainty than the first test case, and it creates only 19.4/20.9 extra A-/C-check slots for the period of 2019–2022, compared with the 20.4/90.4 extra A-/C-check slots used in 2017–2020. Besides, its associated mean average FH is the second-highest for both check type, only after the *myopic* policy.

Both ADP-D and ADP-DS have better performance than the estimation of the airline, in terms of higher mean average FH, fewer mean total checks and mean extra slots for both check types. In fact, the advantage of ADP-DS becomes more notable in this test case. For C-check scheduling and 100 test sample paths, ADP-DS even outperforms ADP-D in all aspects. For A-check scheduling, the extra slots created in the ADP-DS is 75% fewer than in ADP-D. Both methods take just seconds to produce the plan for one day and less than two minutes to produce the schedule for the next month. However, the ADP-DS computation time is 7.5 times as ADP-D due to the Monte Carlo simulations to estimate the cost of performing an A-check action. Looking at the distribution of extra slots in Figure 4, we are aware of the fact that ADP-DS uses fewer than 18 slots in all 100 test sample paths, and in 75% of the test runs, there are less than ten total extra slots. But for ADP-D, the airline may need more than 30 additional slots to cope with uncertainty, and the chance of creating more than ten extra slots is likely to be higher than 90%. Therefore, in the second test case, the ADP-DS is still the best option for the stochastic AMCS. Besides, a Student's t-test also confirms that the results from ADP-D and ADP are significantly different, as shown in Table 11.

Table 11: Student's t-test on the results from ADP-D and ADP-DS for the test case 2019–2022.

t-value	**p-value**	**Degrees of Freedom**	**Pooled Estimate of the Population Standard Deviation**
13.1804	1.6381×10^{-23}	99	6.8283

5.6. Discussion

In the two test cases, we see that the optimal maintenance check schedule from the long-term deterministic AMCS model will likely fail. That is, in the long term, the airline would have to create many additional maintenance slots to cope with the uncertainties from aircraft utilization and maintenance check elapsed time. On the other hand, since it takes only 2–3 seconds for the lookahead ADP methodology to determine the daily optimal maintenance checks, whenever there are changes in maintenance tasks or activities, the airline can use the lookahead ADP methodology to update the maintenance check schedule promptly. Moreover, for each test case, more than 96% of the test runs have the same schedule in the first week, meaning that it is possible for the maintenance planners to update the maintenance check schedule weekly.

Since there is no data about the cost of creating an additional A-/C-check slot, it is impossible to evaluate to what extent reducing aircraft utilization and having maintenance checks earlier is better than creating extra maintenance slots. In the case study, we assumed that creating an additional maintenance slot is costly, more expensive than the cost of anticipating the maintenance check a few flight hours before the end of the interval. Nevertheless, regardless of the real trade-off considered by the user, the lookahead ADP methodology using both deterministic and stochastic forecasts outperforms the *myopic* policy, *thrifty* policy, DP-based methodology described in Deng et al. (2020) and the lookahead ADP methodology itself using only deterministic forecasts.

5.7. Sensitivity Analysis for 2019–2022

This subsection investigates the impact of model parameters of the lookahead ADP methodology on the results of the S-AMCS, for the test case Mar 20$^{\text{th}}$ 2019–Dec 31$^{\text{st}}$ 2022. We are in particular interested in the following aspects:

Q.1 Reducing the number of sample paths for Monte Carlo simulations makes the lookahead ADP methodology faster. How will that affect the results (KPIs)?

Q.2 How much could we improve the KPIs if we increase the number of sample paths for Monte Carlo simulations in the lookahead ADP methodology?

Q.3 If we vary the cost of generating an extra maintenance slot in the lookahead ADP methodology, how will that affect the solutions (KPIs)?

To investigate **Q.1**–**Q.3**, we set up the test scenarios as presented in Table 12. The baseline scenario is the ADP-DS from Table 10. For **Q.1**, if we can still achieve the KPIs within 5% from the ones in the baseline scenario after reducing the number of sample paths for the Monte Carlo simulation, e.g., to 20, it will make the lookahead ADP methodology at least twice faster. In that case, we would suggest using $n_{\text{sample}} = 20$ for the lookahead ADP methodology. For **Q.2**, if we increase the number of sample paths for the Monte Carlo simulation, e.g., from 50 to 80, but achieve no more than 5% improvements in the reduction of extra slots, we suggest using $n_{\text{sample}} = 50$. For **Q.3**, we want to know how many more extra maintenance slots will be created if we reduce the penalty of generating one additional maintenance slot, e.g., from $\lambda = 10^5$ to $\lambda = 100$.

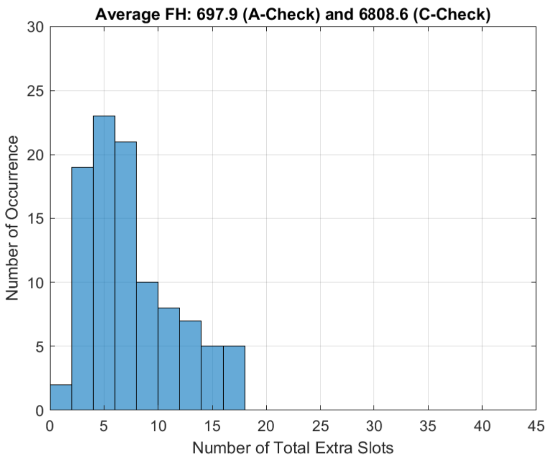
Table 12: Test scenarios for sensitivity analysis.

Test Scenario	**Description**
Scenario 0	Baseline scenario, as pre-computed in Subsection 5.5.
Scenario 1	Conditions from *Scenario 0* and $n_{\text{sample}} = 20$ (240 in total for 12 actions)
Scenario 2	Conditions from *Scenario 0* and $n_{\text{sample}} = 80$ (960 in total for 12 actions)
Scenario 3	Conditions from *Scenario 0* and $\lambda = 100$

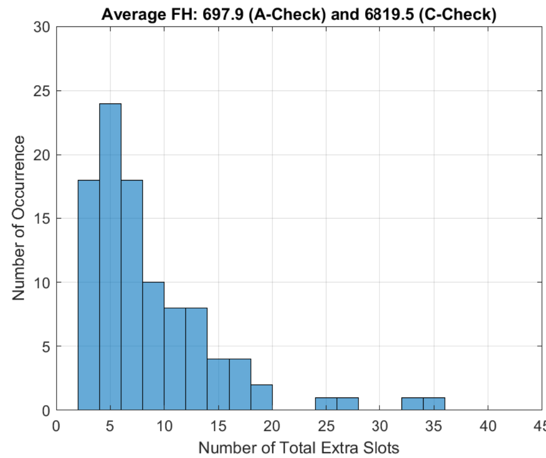
We generate 100 test sample paths for each scenario and apply the lookahead ADP on the S-AMCS. The results are presented in Table 13. For *Scenario 1*, we observe that reducing the number of random sample paths from 50 to 20 in the Monte Carlo simulation increases the mean total extra slots by 1.1 (0.9 for C-check and 0.2 for A-check). At the same time, there is only a minor improvement in aircraft utilization. It also means that the airline needs to create extra slots more frequently than the baseline scenario. Comparing Figure 5a and Figure 5b, we can see the total extra slots scatter between 2 to 35 in *Scenario 1*, one occurrence for 24, one for 26, one for 33 and one for 35 extra slots. It indicates that there would be a 4% chance that the airline may need more than 24 extra slots when we use only 20 sample paths in the Monte Carlo simulation. Since the total number extra slots increase by 15.7% compared with *Scenario 0*, we would not suggest reducing the number of sample paths for the Monte Carlo simulation from $n_{\text{sample}} = 50$ to $n_{\text{sample}} = 20$.

In *Scenario 2*, increasing the number of sample paths for the Monte Carlo simulation from $n_{\text{sample}} = 50$ to $n_{\text{sample}} = 80$ reduces the number of extra slots by 4.2% compared with *Scenario 0*. Although Figure 5c shows that in 76% of the 100 test cases, $n_{\text{sample}} = 80$ results in fewer than 10 extra maintenance slots, only 1% higher than *Scenario 0*, the improvement is not significant since the computation time increases by more than 50%. Hence, we would not suggest increasing the number of sample paths for the Monte Carlo simulation from $n_{\text{sample}} = 50$ to $n_{\text{sample}} = 80$.

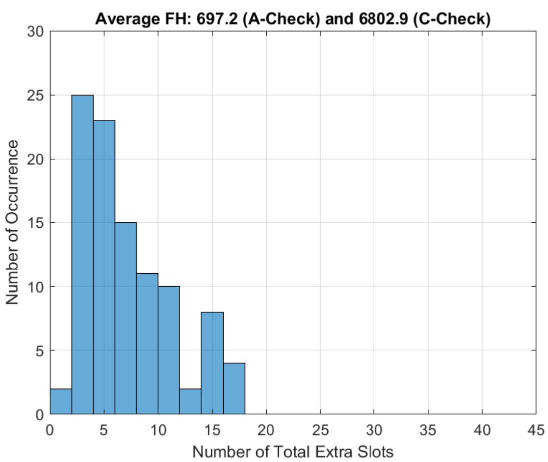
The KPIs of *Scenario 3* indicate that decreasing the cost of creating an extra maintenance slot from 10^5 FH to 100 FH increases the mean total extra slots by 197%, from 7.0 to 20.8 (details can be seen in Figure 5a and 5c). The A-check contributes to most of the extra slots. The approximation of cost function, $\overline{V}_t^{(1)}(s_t)$ in (38), requires that as long as the lookahead ADP methodology predicts an extra C-check slot needed in $[t, t + t_h]$ and there are sufficient C-check slots on the day t, it will choose to perform a C-check. Since there is at most one C-check on the day t, due to a minimum of 3 days between the start dates of two C-checks,



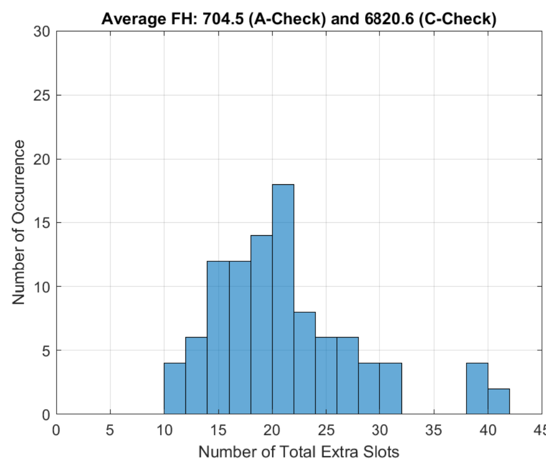
(a) Sensitivity Analysis - Scenario 0



(b) Sensitivity Analysis - Scenario 1



(c) Sensitivity Analysis - Scenario 2



(d) Sensitivity Analysis - Scenario 3

Figure 5: Distributions of total extra slots under different parameters for the lookahead ADP methodology: (a) Distribution of total extra slots of baseline scenario; (b) Distribution of total extra slots when $n_{\text{sample}} = 20$; (c) Distribution of total extra slots when $n_{\text{sample}} = 80$; (d) Distribution of total extra slots when $\lambda = 100$.

Table 13: Sensitivity analysis for the test case March 20th 2019–December 31st 2022 using 100 random sample paths. For each sample path, we use the lookahead ADP methodology to make AMCS decisions for the entire planning horizon.

KPI of 100 Runs (2019-2022)		**Scenario 0**	**Scenario 1**	**Scenario 2**	**Scenario 3**
C-check	Mean Average FH	6808.6	6819.5	6798.0	6820.6
	Mean Extra Slot	4.0	4.9	3.8	5.4
	Mean Total Checks	90.6	90.7	90.8	90.4
A-check	Mean Average FH	697.9	697.9	697.4	704.5
	Mean Extra Slot	3.0	3.2	2.9	15.4
	Mean Total Checks	1019.9	1020.1	1020.6	1010.0
Mean Total Extra Slots		7.0	8.1	6.7	20.8
95% Confidence Interval		[6.19, 7.81]	[6.90, 9.30]	[5.88, 7.52]	[19.45, 22.15]
Mean Merged A- in C-Check		17.6	16.3	17.6	13.5
Computation Time/day [s]		2.63	1.21	4.09	2.63

changing the cost of creating an extra slot λ only has a minor impact on C-check scheduling. On the other hand, since we can perform multiple A-checks on a day, decreasing λ will inevitably increase the number of extra A-check slots (see Figure 5d). Consequently, there is more flexibility in performing aircraft A-check because of the creation of extra A-check slots, and the number of merged A- in C-checks is reduced by 23.3%.

6. Conclusion

This paper proposes a lookahead approximate dynamic programming (ADP) methodology to address the stochastic aircraft maintenance check scheduling (S-AMCS), considering the uncertainty of aircraft daily utilization and maintenance elapsed time. The lookahead ADP methodology consists of a dynamic programming framework and a hybrid lookahead policy with deterministic and stochastic forecasts. The lookahead ADP methodology can provide daily optimal maintenance check decisions and minimize the total unused FH between checks. It increases aircraft availability and reduces the frequency of creating extra maintenance slots in the long term. Eventually, it leads to a significant saving in maintenance operation cost and possibly additional revenue from commercial operation.

The lookahead ADP methodology uses deterministic forecasts first to determine the optimal aircraft C- and D-check actions. Based on the optimal C- and D-check actions, it uses stochastic forecasts to find the best A- and B-check actions. The deterministic forecasts are the estimations of costs of creating extra maintenance slots using the mean aircraft daily utilization and mean maintenance check elapsed time. The stochastic forecasts are the estimations of the costs of generating additional maintenance slots using Monte Carlo simulations. The lookahead ADP methodology determines the daily optimal maintenance check decisions in a matter of seconds, which is suitable for practical day-to-day implementation.

To evaluate the proposed lookahead ADP methodology, we present two case studies using the historical maintenance data of an A320 family fleet from a European airline. On the one hand, in both test cases, we see how that, in the long term, the optimal A- and C-check schedules from the deterministic AMCS create additional maintenance slots to cope with the uncertainty from aircraft utilization and maintenance elapsed time. On the other hand, comparing the KPIs from the maintenance schedule/estimation of the airline and the KPIs from the lookahead ADP methodology, we can infer that the lookahead ADP methodology reduces the total number of letter checks and the number of extra maintenance slots. The reduction of maintenance checks and additional maintenance slots, in the long term, leads to a significant saving in aircraft maintenance costs and generates additional revenue for the airline. The maintenance planners can use the lookahead ADP methodology to update the maintenance check decisions immediately whenever changes occur in the maintenance activities or tasks.

This original and novel study is the first to propose lookahead ADP to make optimal maintenance check decisions daily for the S-AMCS. The lookahead ADP methodology can help maintenance planners react to changes in maintenance activities or tasks faster and promptly update the maintenance check decisions. Maintenance planners can even use the proposed methodology to update short-term schedules (e.g., for the following three days or one week) in 20 seconds once new information is obtained, keeping the letter check schedule optimized for the short term without compromising the long-term feasibility. Besides, it also opens the door for future research on related topics, such as incorporating condition-based maintenance by considering the health prognostics and diagnostics and defining the tasks to be performed within each maintenance check. In this case, we plan the maintenance tasks for each maintenance check according to real-time monitoring rather than fixed intervals. Although this would significantly increase the model complexity, it would extend the S-AMCS to the task level, producing an optimally integrated maintenance check and task execution plan.

Acknowledgments

This research work is part of AIRMES project, which received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 681858. We would like to show gratitude to the associated AIRMES partners for providing their aircraft maintenance data and validation of the proposed methodology and its practical relevance. For more information please visit www.airmes-project.eu.

References

Ackert, S. P. (2010). Basics of Aircraft Maintenance Programs for Financiers. http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers___v1.pdf. (Accessed on September 28, 2017).

AIRBUS (2017). *Airbus A320 Maintenance Planning Document [Private Document]*.

Alfares, H. K. (1999). Aircraft maintenance workforce scheduling: A case study. *Journal of Quality in Maintenance Engineering*, 5(2):78–89.

Başdere, M. and Bilge, U. (2014). Operational aircraft maintenance routing problem with remaining time consideration. *European Journal of Operational Research*, 235(1):315–328.

Boere, N. J. (1977). Air Canada Saves with Aircraft Maintenance Scheduling. *Interfaces*, 7(3):1–13.

Brownlee, J. (2013). What It Was Really Like To Fly During The Golden Age Of Travel. https://www.fastcompany.com/3022215/what-it-was-really-like-to-fly-during-the-golden-age-of-travel. (Accessed on September 27, 2019).

Bruecker, P. D., den Bergh, J. V., Belin, J., and Demeulemeester, E. (2015). A model enhancement heuristic for building robust aircraft maintenancepersonnel rosters with stochastic constraints. *European Journal of Operational Research*, 246(2):661–673.

Cai, C., Wong, C. K., and Heydecker, B. G. (2009). Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies*, 17(5):456–474.

Commercial Aviation Safety Team (2013). Aircraft Design - Original Equipment Manufacturer/Design Approval Holder Continuous Monitoring of Service History and Best Practices Task Force. https://www.skybrary.aero/bookshelf/books/2868.pdf. (Accessed on April 24, 2020).

Deng, Q. (2020). Data set for stochastic aircraft maintenance check scheduling. https://doi.org/10.4121/12993437.v1. Dataset.

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

Eltoukhy, A. E. E., Chan, F. T. S., Chung, S. H., and Qu, T. (2017). Scenario-based Stochastic Framework for Operational Aircraft Maintenance Routing Problem. In *The International Conference on Systems Engineering and Engineering Management*.

Eltoukhy, A. E. E., Wang, Z. X., and Chan, F. T. S. (2018). Joint optimization using a leader-follower stackelberg game for coordinated configuration of stochastic operational aircraft maintenance routing and maintenance staffing. *Computers & Industrial Engineering*, 125:46–68.

European Commission (2015). Airline Maintenance Operations Implementation of an E2E Maintenance Service Architecture and Its Enablers. https://cordis.europa.eu/project/rcn/200486/factsheet/en. (Accessed on September 26, 2019).

Gupta, P., Bazargan, M., and McGrath, R. N. (2003). Simulation Model for Aircraft Line Maintenance Planning. In *Annual Reliability and Maintainability Symposium*.

IATA's Maintenance Cost Task Force (2019). Airline Maintenance Cost Executive Commentary Edition 2019. https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/mctg-fy2018-report-public.pdf. (Accessed on September 11, 2020).

Jorgenson, D. W., McCall, J., and Radner, R. (1966). Optimal Maintenance of Stochastic Failing Equipment. Technical Report. https://www.rand.org/pubs/reports/R437.html. (Accessed on September 27, 2019).

Lagos, C., Delgado, F., and Klapp, M. A. (2020). Dynamic Optimization for Airline Maintenance Operations. *Transportation Science*, 54(4):855–1152.

Li, H. and Womer, N. K. (2015). Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming. *European Journal of Operational Research*, 246:20–33.

Liang, Z., Feng, Y., Zhang, X., Wu, T., and Chaovalitwongse, W. A. (2015). Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B*, 78:238–259.

McGrew, J. S., How, J. P., Williams, B., and Roy, N. (2010). Air-Combat Strategy Using Approximate Dynamic Programming. *Journal of Guidance, Control, and Dynamics*, 33(5):1641–1654.

Medury, A. and Madanat, S. (2013). Incorporating network considerations into pavement management systems: A case for approximate dynamic programming. *Transportation Research Part C: Emerging Technologies*, 33:134–150.

Moudani, W. E. and Mora-Camino, F. (2000). A Dynamic Approach for Aircraft Assignment and Maintenance Scheduling by Airlines. *Journal of Air Transport Management*, 6(4):233–237.

Novoa, C. and Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196:509–515.

Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., and Chryssolouris, G. (2010). An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48(4):604–612.

Powell, W. B. (2011). *Approximate Dynamic Programming - Solving the Curses of Dimensionality*. John Wiley & Sons, Inc., New Jersey.

Rosenberger, J. M., Schaefer, A. J., Goldsman, D., Johnson, E. L., Kleywegt, A. J., and Nemhauser, G. L. (2000). SIMAIR: A Stochastic Model Of Airline Operations. In *Proceedings of the 2000 Winter Simulation Conference*.

Samaranayake, P. and Kiridena, S. (2012). Aircraft maintenance planning and scheduling: an integrated framework. *Journal of Quality in Maintenance Engineering*, 18(4):432–453.

Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problemusing approximate dynamic programming. *European Journal of Operational Research*, 219(3):611–621.

Shaukat, S., Katscher, M., Wu, C.-L., Delgado, F., and Larrain, H. (2020). Aircraft line maintenance scheduling and optimisation. *Journal of Air Transport Management*, 89.

SKYbrary (2019). Maintenance Steering Group-3 (MSG-3). https://www.skybrary.aero/index.php/Maintenance_Steering_Group-3_(MSG-3). (Accessed on September 27, 2019).

Sriram, C. and Haghani, A. (2003). An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment. *Transportation Research Part A*, 37(1):29–48.

Transport Canada (2008). Aeronautical Information Manual, LRA - 2.0 Aircraft Airworthiness, Airworthiness Directives. https://web.archive.org/web/20110417023605/http://www.tc.gc.ca/eng/civilaviation/publications/tp14371-lra-2-0-2566.htm. (Accessed on April 24, 2020).

Tsai, Y.-W. and Gemmill, D. D. (1998). Using tabu search to schedule activities of stochastic resource-constrained projects. *European Journal of Operational Research*, 111(1):129–141.

Vujic, J. L. (2018). Monte Carlo Sampling Methods. http://web.tecnico.ulisboa.pt/~mcasquilho/acad/theo/simul/Vujic.pdf. (Accessed on October 2, 2019).

Zhang, D. and Adelman, D. (2009). An Approximate Dynamic Programming Approach to Network Revenue Management with Customer Choice. *Transportation Science*, 43(3):381–394.

Appendix A. Nomenclature

Indexes:

h_k	Index for a hangar of type k check
i	Aircraft Index
k	Index for maintenance check type
t	Index of calendar day

Parameters:

d_k	Minimum interval between the start dates of two type k checks.
$I^i_{k\text{-DY}}$	Interval of type k check of aircraft i in terms of calendar days (DY)
$I^i_{k\text{-FH}}$	Interval of type k check of aircraft i in terms of flight hours (FH)
$I^i_{k\text{-FC}}$	Interval of type k check of aircraft i in terms of flight cycles (FC)
K	Collection of maintenance check type, $K = \{\text{A-check, B-check, C-check, D-check}\}$
n_{act}	The number of actions on day t
n_{sample}	The number of sample paths generated by Monte Carlo sampling
R^k_{lb}	Lower-bound of *expected remaining utilization* for type k check
t_l	A time period for approximation of future cost for A-/B-check
t_h	A time period for approximation of future cost for C-/D-check
T	Final day in planning horizon
t_0	First day in planning horizon
γ	Discount factor
λ	Daily penalty for having an additional slot for type k check
ξ	A large number to prevent the waste of an available maintenance slot

Exogenous Variables:

$\text{fc}_{t,i}$	Average daily flight cycles usage for aircraft i at day t
$\Delta\text{fc}^{\omega}_{t,i}$	Extra daily FC usage for aircraft i at day t, follows certain distribution
$\text{fh}_{t,i}$	Average daily flight hours usage for aircraft i at day t
$\Delta\text{fh}^{\omega}_{t,i}$	Extra daily FH usage for aircraft i at day t, follows certain distribution
$L_i(y^k_{t,i})$	Mean estimated elapsed time of next check with label $y^k_{t,i}$ of aircraft i
$\Delta L^{\omega}_i\left(y^k_{t,i}\right)$	Extra time needed for the maintenance check labeled as $y^k_{t,i}$, follows certain distribution
W	The set of all sample paths
ω_t	New information that arrives on day t, $\omega_t = \left\{\Delta L^{\omega}_i\left(y^k_{t-1,i}\right),\ \Delta\text{fh}^{\omega}_{t,i},\ \Delta\text{fc}^{\omega}_{t,i}\right\}$

Decision Variables:

x_t^k	A set of actions with respect to type k check on day t, $x_t^k = \left\{\left\{\chi_{t,i}^k\right\}_{i=1}^{N_t}\right\}$
x_t	A set of actions on day t, $x_t = \left\{\left\{\chi_{t,i}^k\right\}_{i=1}^{N_t} \middle\| \sum_{i=1}^{N_t} \chi_{t,i}^k \leq M_t^k\right\}_{k\in K}$
x_t^*	The optimal action among $\{x_t\}$
$X_t(s_t)$	The set of actions of day t from s_t, $X_t = \{\mathscr{X}^\pi(s_t)\}$
$\mathscr{X}^\pi(s_t)$	Scheduling policy function, $\mathscr{X}^\pi(s_t) = \left\{\mathscr{X}_k^\pi\left(s_t^k\right)\right\}_{k\in K}$
$\chi_{t,i}^k$	Binary variable to indicate if aircraft i starts type k check on t

Immediate Reward:

$C_t(s_t, x_t)$	Contribution of choosing action x_t on s_t

State Variables:

A_t	$A_t = \{a_{t,1}, \dots, a_{t,N_t}\}$
$\hat{A}_t$	Post-decision attributes before new information arrives
A_t^k	$A_t^k = \left\{a_{t,1}^k, \dots, a_{t,N_t}^t\right\}$
$a_{t,i}$	The attributes of aircraft i in the beginning of day t
$a_{t,i}^k$	The attributes of aircraft i in the beginning of day t for type k check
$\mathrm{DY}_{t,i}^k$	Total DY of aircraft i in the beginning of day t for type k check
$\mathrm{FC}_{t,i}^k$	Cumulative FC of aircraft i at t since last type k check
$\mathrm{FH}_{t,i}^k$	Cumulative FH of aircraft i at t for type k check
M_t	$M_t = \left\{M_t^k \middle\| k \in K\right\}$
$\hat{M}_t$	$\hat{M}_t = \left\{\hat{M}_t^k \middle\| k \in K\right\}$
M_t^k	Hangar capacity of type k check, $M_t^k = \sum_{h_k} M_{t,h_k}^k$
$\hat{M}_t^k$	$\hat{M}_t^k = M_t^k - \sum_{i=1}^{N_t} \hat{\delta}_{t,i}^k$
M_{t,h_k}^k	Capacity of a hangar h_k specifically for type k check on day t
N_t	Total number of aircraft on day t
$\hat{N}_t$	Post-decision fleet size before new information arrives, $\hat{N}_t = N_t$
$R_{t,i}^k$	Remaining utilization of aircraft i before the next type k check
s_t	Pre-decision state variable, $s_t = \{A_t, M_t, N_t\}$
$\hat{s}_t$	Post-decision state variable before new information arrives
s_t^k	State variable with respect to type k check, $s_t^k = \left\{A_t^k, M_t^k\right\}$
$y_{t,i}^k$	Next maintenance label for type k check of aircraft i on day t
$z_{t,i}^k$	The actual end date of type k check of aircraft i computed on day t
$\hat{z}_{t,i}^k$	The estimated end date of type k check of aircraft i computed on day t

$\delta_{t,i}^{k}$	Binary variable to indicate if aircraft i is undergoing type k check on day t
$\eta_{t,i}^{k}$	Binary variable to indicate if aircraft i needs an extra slot of type k check on day t
Ψ	$\Psi \in \{\text{FH},\ \text{FC}\}$
$\Psi_{t,i}^{k}$	$\Psi_{t,i}^{k} \in \{\text{FH}_{t,i}^{k},\ \text{FC}_{t,i}^{k}\}$
$\psi_{t,i}^{k}$	$\psi_{t,i}^{k} \in \{\text{fh}_{t,i}^{k},\ \text{fc}_{t,i}^{k}\}$

Others:

$\mathcal{S}^{X}(s_t, x_t)$	Transition function from s_t to $\hat{s}_t$, $\hat{s}_t = \mathcal{S}^{X}(s_t, x_t)$ before arrival of new information
$\mathcal{S}^{W}(\hat{s}_t, \omega_t)$	Transition function from $\hat{s}_t$ to s_{t+1}, $s_{t+1} = \mathcal{S}^{W}(\hat{s}_t, \omega_t)$ when the new information is known
$V_t(s_t)$	The value of being in a state s_t
π	Scheduling policy