Delft University of Technology

Optimal Decision Making for Aircraft Maintenance Planning
From Maintenance Check Scheduling to Maintenance Task Allocation

Deng, Q.

DOI
10.4233/uuid:0bfb7a4a-366a-4492-b897-741d3422f9ff

Publication date
2021

Document Version
Final published version

Citation (APA)
Deng, Q. (2021). *Optimal Decision Making for Aircraft Maintenance Planning: From Maintenance Check Scheduling to Maintenance Task Allocation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:0bfb7a4a-366a-4492-b897-741d3422f9ff

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Optimal Decision Making For Aircraft Maintenance Planning

From Maintenance Check Scheduling To Maintenance Task Allocation

Optimal Decision Making For Aircraft Maintenance Planning

From Maintenance Check Scheduling To Maintenance Task Allocation

DISSERTATION

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday 19 April 2021 at 10:00 o'clock

By

Qichen Deng

Master of Science in Mathematics,
KTH Royal Institute of Technology, Sweden
born in Guangzhou, China.

This dissertation has been approved by the promotors.

Composition of the Doctoral Committee:

Rector Magnificus,	chairperson
Prof.dr.ir. M. Mulder,	Delft University of Technology, promotor
Dr.ir. B.F. Lopes dos Santos,	Delft University of Technology, copromotor

Independent Members:	
Prof.dr. R.R. Negenborn	Delft University of Technology
Prof.dr.ir. M. Snellen	Delft University of Technology
Dr. M.T.J. Spaan	Delft University of Technology
Prof.dr.-Ing. V. Gollnick	Technische Universität Hamburg/DLR
Prof.dr. K. Zografos	Lancaster University
Prof.dr.ir. J.M. Hoekstra	Delft University of Technology, reserve member

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. Please visit www.airmes-project.eu for more project information.

Keywords: Scheduling, Decision Support, Aircraft Maintenance, Stochastic Optimization, Dynamic Programming, Bin Packing

Printed by: Ipskamp Printing, Enschede

Front & Back: Q. Deng.

Copyright © 2021 by Q. Deng

ISBN 978-94-6366-398-4

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

The least initial deviation from the truth is multiplied later a thousandfold.

—–Aristotle

Acknowledgements

The journey of my Ph.D. study at Delft University of Technology (TU Delft) began in June 2016. Just like there are lots of sunshine and rain in the Netherlands, I have been through joy, happiness, sadness, and frustrations in the past four and a half years. Finally, it is coming to an end, and the time to earn my Ph.D. degree. At this point, I would like to express my show sincere appreciation to the people who have been accompanying me and supporting me during my Ph.D.:

For the Doctoral Research ...

Prof. Dr. Richard (Ricky) Curran, Dr. Bruno. F. Santos and Dr. Wim J. C. Verhagen for offering me a chance to pursue doctoral research at TU Delft, and Prof. Dr. Max Mulder for being my promotor and helping me in thesis writing.

- Ricky, thank you for bringing me "on board" to the group of Air Transport and Operations (ATO). I appreciate your supervision and the valuable suggestions that you offered as a promotor. Outside doctoral research, I enjoy chatting with you and listening to your music, although I only listen to your music on YouTube and never buy any of your CDs.
- Bruno, thank you for guiding me patiently through my doctoral research, trusting my competence to be a member of the AIRMES project group. I appreciate your constant and careful daily supervision, feedback, suggestions, and fruitful discussions. I have been learning a lot from you while conducting research, both the way you think and the idea you have.
- Wim, thank you for your supports and advice during the AIRMES project. Without you, I would be struggling and wasting more time on the project work. We had been through one of the greatest dangers in life, and we managed to survive in a big fire during our Lisbon trip.
- Prof. Max Mulder, thank you for guiding me at the end of my Ph.D. journey and promoting me to a Doctor of Philosophy. I really appreciate your effort in reading my thesis draft and giving lots of useful comments.

TAP Air Portugal

- Joel Ferreira, Mr. Luís Pimentel de Oliveira, and Mr. Carlos Jorge thank you for your support, feedback, and fruitful discussions on the AIRME project work. I really appreciate the ideas, experience, and insights you brought from the aviation industry, and they contribute to the successful publications of my research work in scientific journals.

AIRBUS

- I want to thank Ms. Daniela Viteri Herrera and Mr. Dang Nguyen-Manh for their support during the AIRMES project. The way you challenge me on the aircraft maintenance planning optimization tool keeps me constantly learning and improving my work and myself.

Colleagues at the ATO Group

- I am always grateful that I had the opportunity to work in the ATO group with amazing colleagues. I want to acknowledge Prof. Dr. Henk A. P. Blom, Prof Dr. Warren Walker, Prof. Dries Visser, Dr. Sander Harjes, Dr. Alexei Sharpanskykh, Dr. Mihaela Mitici, and Ir. Paul Roling, for their assistance in research, your knowledge about the aviation industry is invaluable.

- My fellow Ph.D. colleagues (candidates), Dr. Xiaojia Xhao, Dr. Jeff Newcamp, Dr. Stef Janssen, Dr. Heiko Udluft, Dr. Viwanath Dhaniestty, Dr. Floris Herrema, Dr. Vinh Ho-Huu, Dr. Rui Li, Lennart Scherp, Hemmo Koornneef, Hao Ma, Marie Bieber, Daniel Marta, Matthieu Vert, Chengpeng Jiang, Borrdephong Rattanagraikanakorn, Yalin Li, and Juseong Lee, thank you all for your help during my stay in the ATO group, we really had a great time and lots of fruitful discussions.

- Master Students of ATO, Daniel Coolen, Max Witteman, and Tim van der Weide. It was my honor to be your daily supervisor for your master thesis. We had many discussions about the aircraft maintenance planning problems during your master thesis projects. Your works gave me new insights and inspired me to try different methods to optimize aircraft maintenance planning.

- Special thanks go to Vera van Bragt and Nathalie Zoet for their assistance with the administrative matters. They save me lots of works and time in the office and make my life in the ATO group much earlier.

FRIENDS IN THE NETHERLANDS ...

- Bram Visser, Dhruv Mehta, Agrawal Shruti, Franco Bui Duc, Victor Servando Garcia, Elena Cristiano, and the YOROSHI Budo Group. I want to show my gratitude to all of you who have been helping me and training me all these years. I had lots of fun and lots of pain practicing martial arts with you. It was one of the best experiences of my life.

- Marleen Brouwer, the biggest surprise I have ever had during my Ph.D. study. I really appreciate that we met in the Netherlands, and you were with me during my most difficult time. You brought me joy and happiness to my daily life.

- Bo Han, Feng Lu, Kai Yuan, Lu Li, Lubin Huo, Mei Liu, Mingzhao Zhuo, Shihao Wang, Tianchen Dai, Youwei Wang, Yonghui Huang, Wenwen Sun, and many other Chinese friends. I am very grateful to all of you for your kindness and generosity. You always give me help when I am in trouble.

Relatives and Friends in China ...

I would also like to show my appreciation to all my relatives and friends (especially Sicong Luo, Huiyin Deng, and Lulu Wang) in China for helping my family. You are like the stars in the sky, I may not see you or chat with you every day, but I know you are always there supporting and encouraging me.

Parent ...

Without you, I would not be able to pursue my career or achieve my goals. You are the only ones who show truly selfless, unconditional, and forgiving love. You are my backbone, supporting me no matter where I am and offering me the best you have. You are always with me in the depth of my heart, “traveling” with me all these years to many countries. I cannot express as much as I want in a short paragraph, but I am eternally thankful to you both.

SUMMARY

Aircraft maintenance is the process of overhaul, repair, inspection, or modification of an aircraft or aircraft systems, components, and structures, to keep these in an airworthy condition. Airlines must perform regular maintenance on their fleet to keep their aircraft airworthy and, ultimately, prevent any systems or components failures during commercial operations. Coupled with the rapid growth of the global commercial aircraft fleet, aircraft maintenance demands have increased significantly in the past few decades. Since aviation is a very competitive industry, the growing aircraft maintenance demands and associated operation costs put a huge financial burden on airlines, forcing them to reduce costs while still respecting safety regulations. Therefore, airlines are laying increasing emphasis on planning aircraft maintenance efficiently.

An efficient planning approach for aircraft maintenance is a dual-edged sword. It reduces not only the time and effort of organizing maintenance tasks and coordinating maintenance activities but also increases the time fleet availability for operations and associated revenues. Before introducing wide-body aircraft in the 1970s, airlines used a bottom-up, task-oriented approach to plan aircraft maintenance, as then the commercial fleet sizes were small. Nowadays, most airlines adopt a top-down approach, and first groups the maintenance tasks with the same or similar inspection intervals into a large task block. These, in turn, are commonly divided into four types and labeled as: A-check (every 4–6 months), B-check (every 4–6 months), C-check (every 18–24 months), and D-check (every 6–10 years). After planning the letter checks, airlines further determine the maintenance tasks to be added or removed in each letter check.

This dissertation innovates the aircraft maintenance planning (AMP) process by presenting a comprehensive digital solution. It replaces the current sequential computer-aided manual approach with an integrated scheduling methodology to automate the aircraft maintenance planning process. Given a specific time horizon, it considers all check types together when making the maintenance check decisions and generates the optimal schedules for all letter checks in one comprehensive solution. After that, it plans a long-term (3–5 years) task execution plan based on the optimal maintenance check schedule. These features are integrated into a decision support system (DSS), developed to facilitate aircraft maintenance planning optimization.

The AMP process includes the aircraft maintenance check scheduling (AMCS) and maintenance task allocation. AMCS is the first and also the most important step. The optimal long-term aircraft maintenance check schedule indicates when a particular maintenance task could be performed before it is overdue. This thesis proposes a dynamic programming (DP) based methodology for AMCS optimization. It aims at minimizing the wasted interval between letter checks, considering aircraft type, status, maintenance capacity, and other operational constraints. By achieving this goal, one also limits the number of checks, and with that, reduces maintenance costs.

The allocation of maintenance tasks to letter checks is the second step in the AMP.

After obtaining an optimal aircraft letter check schedule using the proposed DP-based methodology, airlines can add tasks with inspection intervals falling in-between maintenance checks for a given letter check. This thesis formulates the second step of AMP as a time-constrained variable-size bin packing problem (TC-VS-BPP), extending the well-known variable-size bin packing problem (VS-BPP) by adding deadlines, intervals, and repetition of routine tasks. It divides the entire long-term optimal maintenance check schedule into variable-sized bins to which multidimensional tasks are allocated, subject to the available workforce constraints and task deadlines. A constructive heuristic is proposed based on the worst-fit decreasing (WFD) algorithm to address the TC-VS-BPP. The output of the TAP is a long-term task execution plan for each maintenance check.

Although optimizing the AMCS and TAP can provide airlines with a long-term optimal aircraft letter check schedule and an associated task execution plan, it requires complete information on aircraft daily utilization and maintenance check time, excluding future uncertainties. In practice, flight disruptions can impact aircraft utilization, and the routine maintenance tasks can affect maintenance check elapsed time. All these factors may cause deviations from the original maintenance check schedule and task execution plan, requiring the maintenance operators of airlines to regularly adapt the aircraft maintenance check schedule. Following a manual or deterministic scheduling approach may result in insufficient hangar availability at specific moments, requiring the creation of more costly extra maintenance capacity.

This research considers the impact of uncertainty and proposes a lookahead approximate dynamic programming (ADP) methodology for stochastic AMCS optimization. The lookahead ADP methodology adopts a dynamic programming framework, using a hybrid lookahead scheduling policy. The hybrid lookahead scheduling policy makes the optimal decision for heavy aircraft maintenance (C- and D-checks) based on deterministic forecasts and then determines the light maintenance (A- and B-checks) according to stochastic forecasts. The proposed lookahead ADP methodology enables maintenance operators of airlines to make optimal aircraft maintenance check decisions without compromising the long-term AMP efficiency.

Furthermore, this thesis considers the practical application of AMP optimization. A decision support system (DSS) is developed to integrate the deterministic AMCS optimization and associated optimal task allocation. The DSS includes a shift planning function so that the maintenance planners of airlines can use it to plan the work shift and have an overview of the tasks within each work shift and corresponding workload for a short term, i.e., the coming one to two weeks. The DSS was tested and demonstrated in an operational environment, showing its value for real-life implementation. A case study using the fleet maintenance data demonstrates that the DSS is capable of providing an optimal aircraft letter check schedule, a detailed task execution plan, and the work shifts of the coming two weeks in half an hour for a 4-year planning horizon.

In summary, this dissertation proposes a DP-based methodology for long-term deterministic AMCS optimization, a heuristic algorithm for optimal maintenance task allocation of each letter check, a shift planning algorithm to coordinate work shifts and associated tasks, a lookahead ADP for stochastic AMCS optimization, and a DSS to integrate all above AMP functions. From a scientific perspective, this dissertation contributes to the development of a maintenance scheduling methodology, making the optimal main-

tenance decision considering its impact on the future. From an application point of view, this dissertation shows the potential innovation to existing scheduling approaches used by airlines and the feasibility of automating the AMP process. The proposed models, methodologies, and the DSS are demonstrated to be promising in real-life AMP applications and capable of helping airlines make optimal maintenance decisions.

Future research can extend current AMCS and task allocation models to incorporate condition-based maintenance (CBM) by considering the health prognostics and diagnostics and defining the tasks to be performed within each maintenance check. The introduction of CBM to AMP would change the model to plan the maintenance tasks for each maintenance check according to real-time monitoring rather than fixed intervals. This could further increasing aircraft components' life and reduce aircraft maintenance operation costs.

Abbreviations

ADP	Approximate Dynamic Programming
AMCS	Aircraft Maintenance Check Scheduling
AMP	Aircraft Maintenance Planning
AMR	Aircraft Maintenance Routing
BF	Best-Fit
BFD	Best-Fit Decreasing
BPP	Bin Packing Problem
CBM	Condition-Based Maintenance
DP	Dynamic Programming
DSS	Decision Support System
DY	Calendar Days
FF	First-Fit
FFD	First-Fit Decreasing
FH	Flight Hour
FC	Flight Cycle
GUI	Graphical User Interface
IATA	International Air Transport Association
KPI	Key Performance Indicator
MDP	Markov Decision Process
MPD	Maintenance Planning Document
MPP	Maintenance Personnel Planning
MO	Month
MRO	Maintenance, Repair, and Overhaul
MTS	Maintenance Task Scheduling
NF	Next-Fit
NFD	Next-Fit Decreasing
OAMP	Operator Approved Maintenance Program
RUL	Remaining Useful Life
TAP	Task Allocation Problem
TC-VS-BPP	Time-Constrained Variable-Size Bin Packing Problem
VS-BPP	Variable-Size Bin Packing Problem
WF	Worst-Fit
WFD	Worst-Fit Decreasing

Contents

1

INTRODUCTION

1.1. BACKGROUND

Aircraft maintenance planning (AMP) is an intricate problem due to its combinatorial nature and real-life operational constraints. On the one hand, changing weather conditions, flight disruptions, or incidents can affect aircraft utilization, and such impacts cause deviations from the original maintenance plan. On the other hand, additional maintenance needs can also affect the time required for maintenance and changes the maintenance plan. Furthermore, the decision of whether performing a maintenance task on an aircraft today impacts both the use of the aircraft onwards and the need to execute the same task in the future. All these challenges make the AMP generally difficult as the maintenance operators of airlines have to very often adapt the aircraft maintenance schedule to the latest aircraft status and operational constraints.

Regular maintenance inspection prevents aircraft components and systems failures during operations. It involves the overhaul, repair, inspection, or modification of an aircraft or aircraft systems, components, and structures in an airworthy condition [1]. Nowadays, airlines are increasingly interested in planning aircraft maintenance more efficiently since it represents one of the main direct operating costs and plays a vital role in the balance sheet of an airline. In the aviation industry, the spend of global maintenance, repair, and overhaul (MRO) represents 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 [2]. This spending is equivalent to $2.5M per aircraft per year. The savings derived from efficient aircraft maintenance planning can be very substantial: an optimal aircraft maintenance schedule reduces maintenance costs, increases aircraft availability, which in the end, generates additional revenue. Therefore, maintenance operators of the airline aim to allocate maintenance checks on the right aircraft, in the right place, at the right time.

Modern aircraft have thousands of parts, systems, and components that need to be recurrently inspected or replaced. The maintenance planning document (MPD) of an aircraft manufacturer states that each system/component has three usage parameters to indicate its utilization, calendar days (DY), flight hours (FH), and flight cycles (FC):

- DY: One DY is a full 24-hour period,
- FH: FH is the elapsed time between wheel lift off and touch down, and
- FC: One FC is a complete aircraft take-off and landing sequence.

The MPD also defines the inspection interval of a system/component as its maximum usage parameters allowed in commercial operation. Maintenance takes place when a system/component reaches certain DY, FH, or FC thresholds. In real-life applications, maintenance operators usually group maintenance tasks into letter checks, depending on the level of detail: A-check, B-check, C-check, and D-check, and each letter check associates three usage parameters, as shown in Table 1.1.

Table 1.1: Aircraft letter check and corresponding inspection interval [3].

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

The maintenance operators of many airlines plan the aircraft maintenance in three steps, following a top-down approach, according to their experience:

1) Aircraft maintenance check scheduling (AMCS), usually for the future 3–5 years,
2) Task allocation of each maintenance check, usually for the coming year, and
3) Shift planning, usually for the coming 1–2 weeks.

AMCS is a difficult job, especially for an airline with a large, heterogeneous fleet. The main problem associated with current practice is that it is a time-consuming and inefficient process based on the scheduler experience. Maintenance operators often spend several days or weeks scheduling the maintenance check for all aircraft, one after another, according to specific aircraft letter check intervals and the available maintenance resources of the airline. Limited by the (computer-aided) manual planning approach, the maintenance operators usually find a maintenance check schedule for a fleet that is feasible, rather than optimal [4].

Since the maintenance tasks to be executed within each maintenance check are determined based on the aircraft letter schedule, the non-optimal letter check schedule leads to non-optimal task allocation and corresponding work shifts. Besides, the maintenance check schedule is often not capable of being updated quickly due to the lack of an efficient scheduling tool for AMCS in current-day practice. It inevitably decreases aircraft utilization and increases aircraft maintenance costs in the long term. Any change in the maintenance tasks or activities requires the maintenance operators to spend a considerable amount of time shuffling the maintenance checks, re-organizing the associated maintenance tasks and activities, and re-planning the work shifts.

Following the current widely used top-down practice, the optimization of AMP is to find the optimal aircraft maintenance check schedule and the associated task execution plan. It attracts extensive attention from the aviation industry and the scientific community. Both sides started to collaborate to work on this subject by combining the optimization knowledge from academia and experience from airlines. In 2015, an EU project "**Air**line **M**aintenance Operations implementation of an **E**2E Maintenance Service Architecture and Its Enabler**s**" (AIRMES, www.airmes-project.eu) was initiated by the European Commission to optimize end-to-end maintenance activities within an operators' environment [5], led by a major European airline. One of the work packages within AIRMES, which is also the goal of this dissertation, is to develop an optimization framework that can provide maintenance schedule and planning solutions based on operational requirements. In particular, this goal is two-fold:

1) To design methodologies for AMCS optimization and corresponding maintenance task allocation considering the real-life maintenance constraints, and

2) To design a comprehensive tool to reduce the time spent on the AMP process and workload of the maintenance operators.

1.2. RESEARCH QUESTIONS

AMCS is the key to aircraft maintenance planning optimization: the maintenance check indicates the possible periods for the execution of maintenance tasks. The maintenance tasks to be executed within each check further determine the maintenance tools, workforce, and shifts. The main challenge of solving AMCS is to understand the dependency of different check types and estimate the impact of performing a maintenance check on the future. The aircraft A-/B-/C-/D-checks are closely correlated. As described earlier, an aircraft has three usage parameters to indicate each check type's utilization, but all check types are updated with the same daily FH and FC. For example, a C-check lasts 1–4 weeks; whether or not to start a C-check for an aircraft on a particular day can affect the start dates of other check types since this aircraft will not be allocated to any flights during the C-check execution.

On the other hand, the long-term economic and operational benefits of AMCS are often overlooked. In practice, the heavy maintenance (e.g., C- and D-checks) have relatively larger intervals. The status of an aircraft can deviate a lot from expectation before the next C-/D-check, which makes it pointless to spend several days or weeks finding the optimal maintenance check schedule for the entire fleet. Also, there are very few available studies or methods for this subject. For these reasons, some airlines consider a shorter horizon when optimizing the maintenance checks, as then they can see tangible benefits in the nearer future. However, one primary deficiency of this short-term aircraft maintenance planning is that it can be "greedy" and defer all letter checks to a date that is as late as possible. If the maintenance planners of an airline skip one letter check, they may not see any maintenance capacity problem in the coming two or three weeks, yet the maintenance checks overload can happen a few months later. In other words, airlines may get a false impression that the maintenance resources meet the needs of letter checks in a short period, but, as time moves on, the following letter checks can pile up and cause a soaring demand for maintenance in the future, possibly exceeding maintenance capacity.

When the AIRMES project started, the airline partner within AIRMES consortium stated that many European airlines treat the AMCS separately according to letter check type, from heavy to light maintenance. That is, they first focus on scheduling the C-/D-check and then the A-/B-check. The idea of decoupling AMCS, according to check type, significantly reduces the complexity and the time to create a maintenance check schedule. Yet, it neglects the dependency among letter checks, which can easily lead to an either infeasible or a very conservative solution. If a solution is infeasible, the maintenance operators have to repeat a cumbersome process to shuffle the letter checks to make it feasible. If a solution is conservative, it increases the number of maintenance checks, and consequently the maintenance operation costs.

Without an optimal maintenance check schedule, the maintenance operators of airlines are likely to plan the maintenance tasks earlier than the estimated due dates (even in the optimal schedule, the maintenance checks can also start earlier than the estimated due date, but not as far before as in the non-optimal schedule). In the long term, it results in more repetitions of executing the tasks and replacing the systems/components more frequently, reducing the utilization of systems/components and increasing the maintenance operation costs. Moreover, without knowing the optimal maintenance task executions within each letter check, it is also difficult for the maintenance operators to plan the shifts that make full use of the workforce.

The commonly used AMP approaches are mainly depending on the experience of maintenance operators and, in general, inefficient to support airlines in reducing the workloads of maintenance operators or the maintenance operation costs. From a scientific perspective, the biggest challenge in AMP is to integrate different letter check types in the same model formulation and find the unified optimal solution. This requires understanding two important aspects: first, the correlations among different check types, and second, the long-term impact of a maintenance decision for a specific check type on the coming ones of all types. In particular, for the second aspect, the long-term impact of a maintenance check decision on the future is difficult to capture: airlines usually plan the flight schedules only a few weeks beforehand, but the maintenance capacity is predefined. Matching the given maintenance capacity with unknown maintenance check demands has never been done before the AIRMES project.

The challenge in AMP leads to the main research question of this thesis:

> *How to improve the efficiency of maintenance planning for a fleet of heterogeneous aircraft, while considering its longer-term impact on future operations, as well as the uncertainty of daily aircraft utilization and maintenance elapsed time, without compromising safety?*

The main research question is further divided into the following sub-questions:

- How to address the AMCS considering the dependency among different checks?
- What is the optimal maintenance task allocation for each letter check?
- How to address the AMCS considering uncertainties and estimate the long-term impacts of each maintenance check action?

This dissertation investigates the possibility of making optimal aircraft maintenance check decisions by answering the above sub-questions. It shows how the main research question is tackled step by step, from deterministic to stochastic, and from maintenance check scheduling optimization to optimal maintenance task allocation for each check.

1.3. GAP ANALYSIS

Although aircraft maintenance check scheduling (AMCS) is the first and foremost step in aircraft maintenance planning, in general, maintenance scheduling is mandatory not only for aircraft but also for other vehicle types such as bus, train, and ship to maintain vehicles in an operable state.

Bus maintenance scheduling (BMS) can be found in Refs. [6] and [7]. These studies design daily inspection and maintenance schedules for the buses that are due for inspection to minimize the interruptions in the daily bus operating schedule and maximize the utilization of the maintenance facilities. Ref [6] formulated BMS as a classic mixed-integer programming program model and used commercial solver CPLEX to solve the problem; the latter employed a multi-agent system to optimize the bus maintenance schedule heuristically. Similar to buses, trains also undergo daily maintenance inspection. Train maintenance scheduling (TMS) is often coupled within the timetable design, although the primary goal is to optimize train routes, orders, and arrival times at each station. Refs. [8–10] show a few recent studies on TMS. As in BMS, these works adopt similar mixed-integer programming formulations and rely on CPLEX to solve the TMS problems. Unlike BMS and TMS, ship maintenance scheduling (SMS) usually aims to maximize a ship's availability. There are very few studies available for SMS in general. Refs. [11] and [12] model the SMS as constraint satisfaction problems and propose to use genetic algorithms to address SMS.

Compared with BMS, TMS, and SMS, aircraft maintenance scheduling (AMS) is relatively new since traveling by plane was quite expensive and not very popular before the 1970s. The AMS had been using the manual approach for many years. Since the introduction of commercialized wide-body aircraft in the early 1970s, aircraft capacity increased significantly and made flight tickets affordable for millions of travelers. Meanwhile, the AMS has become increasingly difficult due to the emphasis on efficiency and lack of an accurate and timely maintenance scheduling tool. Aircraft manufacturers and airlines started to group maintenance tasks with the same or similar inspection intervals into a large task block, and that was the beginning of using letter checks (A-/B-/C-/D-check). Even so, it still took several weeks for planning personnel to create a maintenance check schedule. Air Canada was aware of this issue and was the first to study aircraft maintenance check scheduling (AMCS). Early in 1977, it presented a priority-based simulation heuristic to produce a feasible maintenance check schedule considering detailed real-life operational constraints [13]. Because of the rapidly changing of aircraft utilization and other unforeseen events, Air Canada did not see the value of using computational power to find an optimal solution that could rapidly become obsolete. The heuristic was very similar to the manual planning approach, shifting conflict checks to earlier time slots until a feasible solution is found, except that it implemented a lower bound of utilization to prevent scheduling checks too often. It reduced the time required to generate a feasible 5-year plan from 3 weeks to a few hours.

Despite the limitations in [13], this is together with Refs. [14] and [15], the only available reference devoted to long-term AMCS. The long-term planning still has not been adequately studied because there was no straightforward method for such a topic, and it was difficult to model the impact of a maintenance check decision. Besides, due to the lack of aircraft maintenance data and daily utilization records (especially the historical maintenance check schedule), it was nearly impossible to formulate the detailed AMCS model. As a result, most research works about aircraft maintenance focus on short-term planning, such as A-/B-check scheduling [16, 17], line maintenance planning [18, 19], or coupled in the literature with the definition of the aircraft routing for the next three to six days of operations [20, 21], that is, assigning each aircraft to a sequence of flight legs (a routing) that allows the aircraft to undergo daily checks [22] or even A-/B-checks [23, 24]. The main reason is that C- and D-checks have intervals of several years, and the benefits of including C-/D-checks in AMCS are only visible in the long term. Airlines usually have higher urgency to monitor and optimize short-term activities, such as aircraft A-/B-check scheduling or routine aircraft inspections, from which they can rapidly see tangible cost savings and profits. In particular, researchers favor the aircraft maintenance routing problem since they have easier access to short-term flight schedules.

Since AMCS is difficult, instead of solving AMCS at the fleet level, some researchers dive into optimizing the maintenance task allocation problem (TAP) for one single aircraft. The idea is to determine the optimal execution of a set of preventive aircraft maintenance tasks so that all of them are performed as close to their estimated due dates as possible. This can be done by combining the maintenance task allocation with aircraft operation to one single problem [4], or focusing on minimizing the overall number of maintenance actions and uniformly distributing the capacity and flying hours over a given time horizon [25], or task clustering [26], or assigning weights on tasks according to ATA code, maintenance interval, zone, and check type, or even using a bottom-up task-oriented approach following the rule of “the most urgent task first” or “the most costly task first/last” [27]. The studies of TAP usually use an aircraft maintenance check schedule (planned by airlines) as an input or know the start date and available workforce beforehand (provided by airlines) so that the researchers do not have to plan the maintenance check schedule own.

Overall, there are very few studies on BMS, TMS, SMS, and AMS/AMCS. Unlike BMS, TMS, and SMS, aircraft maintenance checks have much larger inspection intervals, and a maintenance check schedule is usually planned for a longer-term rather than daily maintenance inspection. An aircraft maintenance check decision can impact aircraft availability, maintenance capacity, or even fleet utilization in the future. As a result, when looking at the AMS/AMCS, one usually considers a much larger time window, e.g., 3–5 years, and the long-term AMCS at fleet level forms a typical large-scale combinatorial problem. To tackle maintenance scheduling, researchers always resort to the solution approaches for general scheduling problems [28], such as an exact method that relies on commercial solver [6, 8–10, 29, 30], or customized methods [7, 11–13, 17, 24]. However, for AMCS, as the fleet size increases, the problem size will increase exponentially, and solving AMCS at the fleet level using an exact method can be computationally expensive. Even for a fleet of 40 aircraft and a 3-year planning horizon, it takes more than half an hour to find the optimal schedule only for one check type [31]. If one includes other check types, it may take hours or days to find the optimal aircraft maintenance check schedule. Furthermore, not even commercial solvers can guarantee a global optimum. Therefore, it is often meaningless to spend lots of time to run a commercial solver to find the global optimal solution. In practice, an aircraft maintenance check schedule that meets the following requirements would be more desirable:

- It is a local optimum,
- It combines all maintenance check types in one single solution, and
- It can be obtained within 10–15 minutes, for a 3–5 year planning horizon.

Once the (local) optimal maintenance check schedule is found, it determines the possible start dates of each maintenance task. The maintenance task allocation can be treated as a bin packing problem. Hence, developing a computationally efficient method to address AMCS has been the main focus of the thesis.

1.4. Research Objective

The research questions and analysis of research gaps provide insights into the formulation of the main research objective:

> *To develop a comprehensive maintenance planning optimization framework, including aircraft maintenance check scheduling and the associated maintenance task allocation, that automates and optimizes the aircraft maintenance planning process without compromising the long-term efficiency.*

The main objective is further divided into two sub-objectives:

O-1 Optimize the aircraft maintenance check schedule and task execution plan

O-2 Automate the aircraft maintenance planning process

For Sub-Objective **O-1**, it covers the following topics:

- Optimize the long-term deterministic aircraft maintenance check schedule,
- Optimize the task execution plan for each maintenance check, and
- Optimize the aircraft maintenance check decision considering uncertainties.

The idea of including uncertainties in AMCS is that the deviation of actual maintenance check elapsed time from planning can impact the aircraft utilization of future and the following checks of the entire fleet, which may result in an update of all current or upcoming maintenance decisions of the next few days or even weeks. Changing maintenance decisions often hinders the planning efficiency and increases the workload of maintenance operators of airlines since they have to re-organize the maintenance tools and coordinate the workforce. Moreover, changing aircraft maintenance check decisions can affect the flight plan and lead to extra work for the staff of the operations center since they may need to re-design the flight schedule or adjust the flight legs and cabin crew.

Achieving **O-1** would allow maintenance operators of airlines to obtain a consistent optimal aircraft maintenance planning solution, from the long-term maintenance schedules and task execution plans to short-term maintenance check decision updates. For **O-2**, the key is to develop a decision-making framework integrating both AMCS and maintenance task allocation under the same platform. A prominent feature of the model framework is that the user only needs to load input data via an interface, and then the model framework automatically generates an optimal solution.

1.5. RESEARCH METHODOLOGY

The lack of efficient optimization algorithms for aircraft maintenance check scheduling (AMCS) is the main difficulty in improving aircraft maintenance planning (AMP), which further hinders the associated maintenance task allocation since the task allocation requires the start dates of each maintenance check. The main challenges in addressing AMCS are:

- **C.1** No maintenance planning documents and maintenance check and task execution data are available for researchers since they are usually confidential for the airlines.
- **C.2** No information about the maintenance capacity (e.g., number of hangars and workforce composition) or detailed maintenance operational constraints exists.
- **C.3** No literature exists about current-day AMCS optimization, especially for heavy maintenance checks (C-/D-checks); researchers have to explore the solution approaches themselves.
- **C.4** No aircraft maintenance cost data are available to model the impact of a maintenance check decision properly.
- **C.5** No maintenance check schedules from airlines are available for validation purposes since these are also confidential for the airlines.
- **C.6** No information about how often the maintenance operators update an aircraft maintenance check schedule exists.
- **C.7** It is difficult to collect historical aircraft utilization data to test the robustness of a novel maintenance check schedule.

The research methodology of this thesis aims to address these above challenges. As shown in Figure 1.1, the research methodology is divided into four phases.

1.5.1. PHASE-I PREPARATION

This phase is to address challenges **C.1** and **C.2**. It aims to understand AMCS and the associated task allocation problem. The first step was to study the historical maintenance check schedule and task execution plan, and the MPD from aircraft manufacturers. In this way, we learn how airlines address AMCS and the associated TAP in practice and can obtain an overview of the workload of each check/task and task execution sequence. The second step was to review the literature about aircraft maintenance and maintenance related topics. This step provides some insights into the modeling of aircraft maintenance

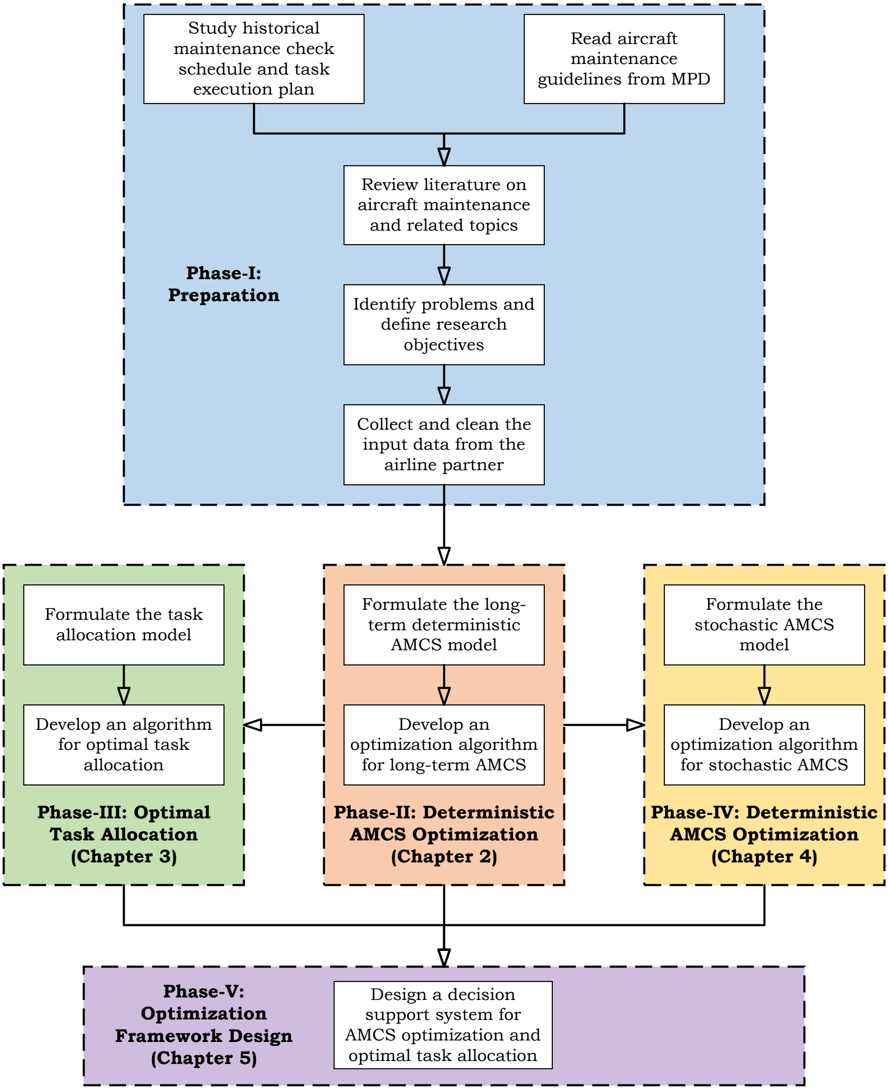


Figure 1.1: The research methodology for aircraft maintenance planning optimization.

problems. The third step is to identify which type of optimization problems AMCS and TAP belong to. In the AMCS, the fleet status of a day partly depends on the previous maintenance check actions and partly depends on the aircraft utilization. Every day, the maintenance operators face the problem of deciding whether to send an aircraft for a maintenance check or let it fly for another day, and each decision is associated with a cost. The state of the fleet at any time is characterized by the usage parameters of A-, B-, C-, and D-check. The AMCS problem is to determine the policy which minimizes the expected total cost. Once a maintenance check schedule is found, the TAP then determines the optimal start dates for a set of maintenance tasks. The goal is to minimize the total cost for maintenance execution while ensuring that each task is executed before its expected due date. The last step of Phase-I is to collect and clean the input data for both AMCS and TAP.

1.5.2. PHASE-II DETERMINISTIC AMCS OPTIMIZATION

Phase-II is the key to AMP optimization and to tackle challenges C.3–C.5. In this phase, the deterministic AMCS problem is formulated, including defining the proper objective function and model the operational constraints in detail. After that, it continues to develop the methodology for deterministic AMCS optimization, based on the findings from the literature review in Phase-I. Two research directions are derived from deterministic AMCS optimization:

(i) Optimal task allocation of each maintenance check (Phase-III);

(ii) Stochastic AMCS optimization (Phase-IV).

1.5.3. PHASE-III OPTIMAL TASK ALLOCATION

The optimal task allocation (in the context of AMP) is the follow-up of deterministic AMCS optimization. The Phase-III is to model the TAP, e.g., formulate the objective functions and constraints, based on the optimal maintenance check schedule. After that, it continues to the development of the optimization algorithm for the TAP.

1.5.4. PHASE-IV STOCHASTIC AMCS OPTIMIZATION

The stochastic AMCS optimization is derived from the deterministic AMCS optimization. Phase-IV addresses challenges C.6 and C.7, by including stochastic elements into the AMCS model, such as the uncertainties in aircraft daily FH, FC, and maintenance elapsed time. This phase aimed to develop a fast and efficient approach to re-compute the optimal maintenance check actions for the short term, without compromising the efficiency of future decisions.

1.5.5. PHASE-V OPTIMIZATION FRAMEWORK DESIGN

Phase-V aimed to design a decision support framework that integrates deterministic AMCS optimization, optimal task allocation, and stochastic AMCS optimization. Besides, maintenance shift planning is also one of the main focuses. The decision support framework for AMP is designed to facilitate maintenance planners of airlines making optimal maintenance decisions, from long-term AMCS and task allocation to short-term shift planning and workload estimation.

1.6. SCIENTIFIC CONTRIBUTIONS

Overall, the contribution of this dissertation is two-fold. First and foremost, it contributes to the methodologies design for aircraft maintenance planning. In particular, it presents methodologies for aircraft maintenance check scheduling (AMCS) optimization and optimal maintenance tasks allocation:

- This dissertation is the first to optimize the long-term deterministic AMCS. It proposes a priority solution to reduce aircraft selection possibilities for maintenance checks, a thrifty algorithm to infer the impact of a maintenance check decision, and a discretization and state aggregation scheme to reduce the outcome space. The corresponding scientific contribution led to the following publication:

  Deng, **Q**., Santos, B. F., and Curren, R. (2020). A Practical Dynamic Programming based Methodology for Aircraft Maintenance Check Scheduling Optimization. *European Journal of Operational Research*, 281(2), 256-273.

- This dissertation is the first to optimize the long-term aircraft maintenance task allocation. It proposes an optimal algorithm that allocates maintenance tasks to each aircraft letter check in a reasonable and stable computation time, regardless of how limited maintenance resources are. The corresponding research led to the following working paper:

  Witteman, M., **Deng**, **Q**., and Santos, B. F. (2021). A Bin Packing Approach to Solve the Aircraft Maintenance Task Allocation Problem. *European Journal of Operational Research* (DOI: https://doi.org/10.1016/j.ejor.2021.01.027).

- This dissertation is the first to include uncertainty in AMCS and solve the stochastic AMCS optimization problem. It presents a methodology for re-compute short-term AMCS decisions without future AMCS efficiency. The development and validation of optimization methodology led to the following working paper:

  Deng, **Q**. and Santos, B. F. (2021). Lookahead Approximate Dynamic Programming for Stochastic Aircraft Maintenance Check Scheduling Optimization. *European Journal of Operational Research*, submitted.

Besides the scientific innovation, this dissertation also contributes to the practical application of aircraft maintenance planning (AMP), especially in the improvement of AMP efficiency and reducing the workload of AMP personnel:

- It develops the first decision support system (DSS) to optimize the long-term aircraft maintenance check schedule, task allocation, and short-term work shift planning. The DSS is capable of computing a 3-year, comprehensive, optimal aircraft maintenance plan within half an hour. The development of the DSS resulted in the following working paper:

  Deng, **Q**. and Santos, B. F., and Verhagen, W. J. C. (2021). A Novel Decision Support System for Optimizing Aircraft Maintenance Check Schedule and Task Allocation. *Decision Support Systems* (DOI: https://doi.org/10.1016/j.dss.2021.113545).

1

1.7. Overview of Dissertation

For the ease of navigation, this thesis is divided into six chapters. Chapters 2—5 correspond to Phase-II—V of the research methodology. Each of these chapters also includes a part of the literature study performed in Phase-I. Chapter 2 is the core of this dissertation. It presents the model formulation for deterministic AMCS optimization and an associated dynamic programming (DP) based methodology. After the readers understand the purpose about the deterministic AMCS model from 2, they can continue with Chapter 3 and Chapter 4. These two chapters discuss two different research directions derived from deterministic AMCS. Chapter 3 describes the model formulation and a heuristic algorithm for optimal task allocation for each maintenance check. Chapter 4 presents the model formulation and a lookahead approximate dynamic programming (ADP) methodology for the stochastic AMCS. Chapter 5 depicts a decision support system (DSS) for AMP optimization that integrates the models from Chapter 2—Chapter 4. The last chapter summarizes this thesis with concluding remarks and gives an outlook on future work.

References

[1] Minister of Justice, *Canadian Aviation Regulations 2012-1, Part I - General Provisions, Subpart 1 - Interpretation,* (2012), (Accessed on September 28, 2017).

[2] IATA's Maintenance Cost Task Force, *Airline Maintenance Cost Executive Commentary Edition 2019,* (2019), (Accessed on September 11, 2020).

[3] S. P. Ackert, *Basics of Aircraft Maintenance Programs for Financiers,* (2010), (Accessed on September 28, 2017).

[4] C. Van Buskirk, B. Dawant, G. Karsai, J. Sprinkle, G. Szokoli, and R. Currer, *Computer-aided aircraft maintenance scheduling*, Tech. Rep. (Institute for Software-Integrated Systems, 2002).

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

[6] A. Haghani and Y. Shafahi, *Bus maintenance systems and maintenance scheduling: model formulations and solutions,* Transportation Research Part A: Policy and Practice **36**, 453 (2002).

[7] R. Zhou, B. Fox, H. P. Lee, and A. Y. C. Nee, *Bus maintenance scheduling using multi-agent systems,* Engineering Applications of Artificial Intelligence **17**, 623 (2004).

[8] X. Luan, J. Miao, L. Meng, F. Corman, and G. Lodewijks, *Integrated optimization on train scheduling and preventive maintenance time slots planning,* Transportation Research Part C: Emerging Technologies **80**, 329 (2017).

[9] B. Lin, J. Wu, R. Lin, J. Wang, H. Wang, and X. Zhang, *Optimization of high-level preventive maintenance scheduling for high-speed trains,* Reliability Engineering and System Safety **183**, 261 (2019).

[10] C. Zhang, Y. Gao, L. Yang, and U. K. Z. iyou Gao, *Integrated optimization of train scheduling and maintenance planning on high-speed railway corridors,* Omega **87**, 86 (2019).

[11] S. Deris, S. Omatu, H. Ohta, S. Kutar, and P. A. Samat, *Application of a Hybrid Genetic Algorithm to Ship Maintenance Scheduling,* IFAC Proceedings Volumes **30**, 65 (1997).

[12] S. Deris, S. Omatu, H. Ohta, and L. S. K. P. A. Samat, *Ship maintenance scheduling by genetic algorithm andconstraint-based reasoning,* European Journal of Operational Research **112**, 489 (1999).

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

[14] M. Etschmaier and P. Franke, *Long-Term Scheduling of Aircraft Overhauls,* in *AGIFORS Symposium* (Broadway, Great Britain, 1969).

[15] H. Bauer-Stämpfli, *Near Optimal Long-Term Scheduling of Aircraft Overhauls by Dynamic Programming,* in *AGIFORS Symposium* (Benalmadena, Spain, 1971).

[16] C. Sriram and A. Haghani, *An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment,* Transportation Research Part A **37**, 29 (2003).

[17] C. Lagos, F. Delgado, and M. A. Klapp, *Dynamic Optimization for Airline Maintenance Operations,* Transportation Science **54**, 855 (2020).

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

[19] S. Shaukat, M. Katscher, C.-L. Wu, F. Delgado, and H. Larrain, *Aircraft line maintenance scheduling and optimisation,* Journal of Air Transport Management **89** (2020), https://doi.org/10.1016/j.jairtraman.2020.101914.

[20] M. Başdere and U. Bilge, *Operational aircraft maintenance routing problem with remaining time consideration,* European Journal of Operational Research **235**, 315 (2014).

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

[22] P. Belobaba, A. Odoni, and C. Barnhart, *Global Airline Industry* (John Wiley and Sons, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2009).

[23] T. A. Feo and J. F. Bard, *Flight Scheduling and Maintenance Base Planning,* Management Science **35**, 1415 (1989).

1

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

[25] A. Steiner, *A Heuristic Method for Aircraft Maintenance Scheduling under Various Constraints*, in *6th Swiss Transport Research Conference* (Monte Verità, Ascona, 2006).

[26] A. K. Muchiri and K. Smit, *Application of Maintenance Interval De-Escalation in Base Maintenance Planning Optimization*, Enterprise Risk Management **1** (2009), https://doi.org/10.5296/erm.v1i2.179.

[27] N. Hölzel, C. Schröder, T. Schilling, and V. Gollnick, *A Maintenance Packaging and Scheduling Optimization Method for Future Aircraft*, in *Air Transport and Operations Symposium* (2012).

[28] S. O. Duffuaa and K. S. Al-Sultan, *Mathematical programming approaches for the management of maintenance planning and scheduling*, Journal of Quality in Maintenance Engineering **3**, 163 (1997).

[29] H. Go, J.-S. Kim, and D.-H. Lee, *Operation and Preventive Maintenance Scheduling for Containerships: Mathematical Model and Solution Algorithm*, European Journal of Operational Research **229**, 626 (2013).

[30] A. Kiefera, M. Schildeb, and K. F. Doerner, *Scheduling of Maintenance Work of a Large-Scale Tramway Network*, European Journal of Operational Research **270**, 1158 (2018).

[31] T. M. J. van der Weide, *Long-term C-Check scheduling for a fleet of heterogeneous aircraft under uncertainty*, Master's thesis (2020).

2

Deterministic Aircraft Maintenance Check Scheduling Optimization

The study in this chapter aims to model the long-term, deterministic aircraft maintenance check scheduling (AMCS) and present the corresponding solution—a practical dynamic programming (DP) based methodology. The deterministic AMCS model formulation considers aircraft type, fleet status, maintenance capacity, and other detailed operational constraints. The DP based methodology adopts the idea of forward induction, incorporating a maintenance priority solution to reduce the action space, a discretization and state aggregation strategy to trim the outcome space, and a thrifty algorithm to estimate the consequence of performing a maintenance check action. It is the first methodology to optimize AMCS considering multiple check types in one single problem. The deterministic AMCS model and corresponding solution are applied to a real-life case study and validated in collaboration with one of the major European airlines.

The content of this chapter is based on the following research article:

Deng, Q., Santos, B. F., and Curren, R. (2020). A Practical Dynamic Programming based Methodology for Aircraft Maintenance Check Scheduling Optimization. *European Journal of Operational Research*, 281(2), 256-273. To cite this article, please use the DOI https://doi.org/10.1016/j.ejor.2019.08.025.

2.1. INTRODUCTION

Aircraft maintenance is the overhaul, repair, inspection, or modification of an aircraft or aircraft systems, components, and structures in an airworthy condition [1]. Regular maintenance prevents aircraft components and systems failures during operations. It takes place when an aircraft undergoes certain flight hours, flight cycles, or calendar months. There are three major types of maintenance: A-check, B-check[1], C-check and D-check. A typical A-check includes inspection of the interior or exterior of the airplane with selected areas opened (e.g., checking and servicing oil, filter replacement, and lubrication) [2]; they are performed approximately every 2 to 3 months. The C-check requires a thorough inspection of individual systems and components for serviceability and function; it is planned within an interval of 18 to 24 months. The D-check (a.k.a Structural Check) uncovers the airframe, supporting structure, and wings to inspect the most structurally significant items; it is carried out every 6 to 10 years. Many airlines merge D-check into C-check and label it as a heavy C-check. During a C-check or D-check, the aircraft has to be grounded for several weeks and removed from the revenue schedule. For the first time, this chapter optimizes the long-term AMCS, integrating multiple check types in the same problem. We call this problem the aircraft maintenance check scheduling problem, or for short, the AMCS problem.

Scheduling the maintenance inspection for a large heterogeneous fleet is generally a demanding and complex problem. In practice, the aircraft maintenance schedules are usually prepared according to the experience of maintenance operators. The main problem associated with such a planning approach is that it is time-consuming and can result in poor solutions. For a large fleet, the maintenance operators need to spend several days or weeks planning the maintenance checks one after another according to individual aircraft inspection intervals and maintenance resources of the airline. If conflict maintenance checks occur, the maintenance operator needs to adjust the schedule, constantly moving checks to earlier or later time slots until a feasible schedule is found. Limited by the manual planning approach, the goal is usually to find a feasible maintenance schedule for a fleet instead of an optimal one [3]. As a result, the traditional manual maintenance planning approach inevitably decreases aircraft utilization and leads to more maintenance checks in the long term, increasing aircraft maintenance costs. Therefore, an optimized long-term maintenance schedule reduces the number of maintenance checks and increases aircraft availability, the saving derived from efficient maintenance planning can be very substantial.

Nowadays, airlines are laying increasing emphasis on improving their aircraft availability and planning their maintenance in a more efficient way. Aircraft maintenance represents one of the main direct operating costs and plays an important role on the balance sheet of an airline. According to [4], 9%–10% of the total cost of an airline goes to aircraft maintenance. This was equivalent to $295M on average per year per airline [5]. The long-term economic and operational benefits of adopting a more efficient approach are clear; a typical C-check of A320 family may cost $150k—$350k [2], an A-check cost around $10k—$15k, while an additional day on operation may represent $75k—$120k of commercial revenue (depending on the utilization level of aircraft). However, the chal-

[1]B-checks are rarely mentioned in practice. The tasks included in B-checks are commonly incorporated into successive A-checks

lenge is that the maintenance check scheduling problems are several correlated combinatorial problems. The decision to schedule or not schedule one maintenance check on an aircraft today impacts the utilization of the aircraft onwards and, therefore, on the need to perform maintenance checks in the future. This kind of problem is hard to solve and is often addressed by heuristics or algorithms [6].

This chapter proposes a dynamic programming (DP) based methodology to solve the AMCS problem. The main contributions can be summarized in the following:

1) Methodology:
   - An innovative and tractable DP-based model formulation is presented, suitable to solve real-life, large scale scheduling problems.
   - A thrifty algorithm is used to infer future implications of an action taken at the current stage.
2) Practicality
   - The optimization takes the inspection interval of different check types and detailed operation constraints into consideration.
   - It takes less than 15 min to optimize the 4-year A- and C-check schedule for more than 40 aircraft, rather than days or weeks.
3) Application
   - For the first time, the long-term AMCS problem is formulated and optimized by a single algorithm.
   - The formulation is flexible, and other maintenance events can be easily included in the proposed model, such as landing gear maintenance or cabin modification.

The outline of this chapter is as follows: Section 2.2 reviews the literature about aircraft maintenance planning and solution techniques for scheduling problems. The aircraft maintenance constraints and AMCS problem formulation are presented in Section 2.3. The DP based methodology for AMCS optimization is discussed in detail in Section 2.5. Section 2.6 describes the case study from a European airline. The last section summarizes the research with concluding remarks and gives an outlook on future work.

2.2. LITERATURE OVERVIEW

Aircraft maintenance check scheduling (AMCS) has been relying on the manual planning approach for many years. Since the introduction of commercialized wide-body aircraft in the early 1970s, AMCS has become increasingly difficult due to the emphasis on efficiency and lack of an accurate and timely maintenance scheduling tool. It usually took several weeks for planning personnel to create a maintenance schedule [7]. Air Canada was aware of this issue in the 1970s and developed an aircraft maintenance operations simulation model (AMOS) to improve maintenance efficiency and reduce labor and material cost [7]. The AMOS tool formulated the AMCS as a discrete integer programming problem. According to the author, the problem constraints in AMOS included workforce, public holidays, and summer period (when no maintenance was allowed), contractual maintenance duties for other airlines, reliability, and required inspection intervals from the maintenance planning document (MPD). Several assumptions had been made in AMOS: each maintenance event ties up only one hangar/slot; the minimum time unit is one calendar day; aircraft is aged by daily flight hours; maintenance events can be postponed from the desired due date within a certain tolerance. Although AMOS works for both A- and C-checks, intending to minimize total unused flight hours between two successive C-checks, a priority-based simulation heuristic is used to produce (good) feasible solutions. The author claims that neither mixed-integer programming nor dynamic programming is deemed suitable to Air Canada's environment. Furthermore, due to the rapidly changing of aircraft utilization and other unforeseen events, the author did not see the value of using computational power to find an optimal solution that could rapidly become obsolete. Therefore, a simulation-based approach was adopted in AMOS, in which the user is the one that chooses the best solution. The process is very similar to the manual planning approach, shifting conflict checks to earlier time slots until a feasible solution is found, except that a lower bound of utilization was implemented to prevent scheduling checks too often. Still, the main contribution of AMOS was to propose a systematic maintenance scheduling approach that could reduce the time required to generate a 5-year plan from 3 weeks to a few hours [7].

Despite the limitations in AMOS, this is together with Ref. [8] and [9], the only available reference devoted to long-term aircraft maintenance scheduling. The long-term planning still has not been adequately studied. In fact, most research works about the aircraft maintenance topic focus on short-term aircraft maintenance routing, that is, ensuring that each aircraft is assigned to a sequence of flight legs (a routing) that allows the aircraft to undergo daily checks, which are needed every two to four days [10]. The main reason is that aircraft maintenance checks have intervals of several months/years, and the benefits of an optimal schedule are only visible in the long term. Airlines usually have higher urgency to monitor and optimize short-term activities, such as aircraft routing and routine aircraft inspections, from which they can rapidly see tangible cost savings and profits.

Ref. [11] is one of the first works to address the aircraft maintenance routing problem. It primarily focuses on the flight schedule design and incorporates the maintenance requirement as part of the constraints. A homogeneous fleet and fixed time intervals between maintenance checks were considered for simplicity. Only A-checks are considered in this work since C-checks are spaced at relatively large time intervals. The planned flight schedule minimizes the total maintenance cost and also determines the maintenance base for the aircraft, which starts and ends in the same city. The problem is formulated as a min-cost, multi-commodity flow network with integer constraints. Column generation is applied to obtain an optimized solution. Although the main purpose of [11] is to design a flight schedule, it is considered as a significant step in maintenance planning. Several authors have followed this path and continued the research on aircraft maintenance routing, such as [12], [13], [14] and [15]. For example, Ref. [12] proposed a hybrid dynamic programming (DP) approach, which recursively searched for the best maintenance schedule, followed by a greedy algorithm to solve the sequential maintenance schedule problem. This approach was developed specifically as an on-line fleet operations management decision support system, focused on providing improved daily aircraft assignment solutions based on a given aircraft maintenance check schedule.

Ref. [16] is one rare example where the focus has been shifted from flight schedule design to maintenance scheduling. The authors proposed a mixed random search and depth first search heuristic to minimize the total costs of A- and B-checks and inappropriate aircraft assignments. Unlike other research works that emphasize flight schedule design and consider aircraft maintenance as a constraint, the flight schedule is given as input. The goal is to determine when, where, and what type of maintenance check an aircraft should undergo. Although the main focus is on maintenance scheduling, this research is still considered as an extension of [11]. C-check scheduling has not been considered since long-term flight schedules are unknown.

Instead of scheduling letter checks (A-, C- and structural/D-check), an alternative that reduces maintenance costs and generates profits in the short-term is aircraft maintenance task scheduling. This approach reverses the conventional top-down stereotyped planning. It schedules tasks individually, which gives flexibility to maintenance operators to execute the tasks at the most appropriate time [17, 18]. The task-oriented planning concept and its application are illustrated in [19]. The case study claims that more than $4M can be saved over 72 days compared with the rigid letter checks. However, there is little information about the influence of fleet size, maintenance capacity, and algorithm computation time among the works concerning task-oriented planning approaches. Since an aircraft can have about 2000-3000 maintenance tasks, the practicality of applying a task-oriented approach to planning a long-term maintenance schedule for a large fleet remains questionable.

In general, the literature on long-term maintenance scheduling is limited, and researchers often resort to solution techniques from more general scheduling problems. Ref. [20] summarizes a list of objective functions, models, and optimization methods of scheduling problems. Since scheduling problems usually involve integer decision variables and linear constraints, the most common approaches to such mixed-integer linear programming problems are heuristics, which rely heavily on commercial solver such as CPLEX [21, 22]; the other alternative is dynamic programming (DP).

DP was proposed in the 1950s by Bellman, referring specifically to nesting smaller decision problems inside larger decisions [23]. It divides a large and complicated problem into stages and states. The smaller sub-problems within each state are solved faster than the initial problem, and the optimal solution can be retrieved by examining the solutions from all sub-problems. DP was initially applied to single-machine production [24–26] or single-machine maintenance scheduling problems [27]. For example, the work from [24], which minimizes the total cost of producing different items from a single machine, is considered to be one of the first to motivate the application of DP on scheduling problems. However, it assumes that only one unit can be produced at a time and the demand rate of units is constant.

As the development of DP, the application has been gradually extended from single-machine scheduling to multiple-machine scheduling. Most of the DP applications on multiple machines are related to power generation. One of the examples can be found in [28]. It presented a study to optimize the cost of multiple reservoir systems over a long period considering the uncertainties of water inflow and equipment outage. Even though piecewise linear functions are introduced in the solution process to estimate the future operation cost (this avoids recursively computing the actual future operating cost), the application is still limited to low-dimensional problems, namely, a small number of reservoirs [29]. When the number of reservoirs increases, the number of decisions, i.e., how much water should be kept in each reservoir, increases exponentially.

Several conclusions can be drawn from the review of the literature. First of all, Ref. [7] is the only available reference for long-term AMCS, although no optimization technique is implemented. Secondly, the long-term AMCS forms a typical large-scale combinatorial problem, but there is no standard approach or exact algorithm for such a problem type. Thirdly, aircraft A- and C-check scheduling on a fleet aggregate level is analogous to multiple unit scheduling, which can be treated with the similar formulation and solution techniques. Fourthly, DP is capable of dealing with small-scale mixed-integer/combinatorial problems, but the classic DP is not applicable to large-scale problems. The rest of this chapter will present a DP based methodology to tackle the long-term aircraft maintenance check scheduling optimization.

2.3. PROBLEM FORMULATION

This section formulates the AMCS problem, adopting the DP framework. It starts with an introduction of inspection intervals (2.3.1), followed by a list of assumptions (2.3.2). After that it explains the maintenance capacity and some common operational constraints (2.4.1). The formulation of the AMCS problem is then described, divided into decision space (2.4.2), definition of state (2.4.3), state transition (2.4.4), constraints formulation (2.4.5) and the objective function (2.4.6). The final subsection summarizes the optimization model formulation.

2.3.1. MAINTENANCE INSPECTION INTERVAL

In the aviation industry, aircraft are aged by daily utilization with respect to 3 different usage parameters, calendar day (DY), flight hours (FH), and flight cycles (FC). One DY is a full 24 hours period; FH refers to the elapsed time between wheel lift off and touch down; and an FC is defined by a complete take-off and landing sequence. The inspection interval reflects the maximum usage parameters allowed in operation. For example, the maintenance planning document (MDP) of the AIRBUS A320 family [30] defines that a C-check interval corresponds to 730 DY, 7500 FH or 5000 FC; and 120 DY, 750 FH or 750 FC for the A-check.

After a maintenance check, the corresponding three usage parameters are set to 0, and a maintenance cycle is concluded. These maintenance cycles are associated with labels, referring to different task packages (i.e., A1, A2, A3,... for the A-check and C1, C2,... for the C-check). The A-check program is commonly divided into 4 cycles, in which A1 has similar task packages as A5, while A2 has similar task packages as A6, and so forth. The C-check program has 12 cycles and consists of continuous C-checks, whereby every three checks (i.e., C3, C6, ...), there is a heavier check incorporating tasks from D-checks.

The aircraft MPD also includes an inspection interval tolerance. This tolerance allows operators to fit the maintenance schedule around maintenance capacity, maintenance operational constraints, and commercial operation demands. However, if tolerance is used in one maintenance cycle, the amount of DY, FH, and FC used from the tolerance needs to be deducted from the maximum usage parameter values for the next cycle. This guarantees that the maximum usage parameters are verified in the long term. The inspection interval tolerance should not be included as a planning option, but it is commonly used in practice to accommodate deviations from the initial schedule.

Although having different usage parameters, different check types within AMCS are correlated due to two reasons: the first is that when an aircraft is performing a letter check (e.g., A-check), it will be grounded, and the usage parameters of other check types (B-, C- and D-checks) are not altered (i.e., the daily utilization of these parameters is equal to 0). The second reason is that, depending on the usage parameters for the A-check, it could be beneficial for the airline to merge the A-check within a C-/D-check. This has the advantage of performing the A-checks without necessarily increasing the C-/D-check duration and without using an A-check slot. On the other hand, to anticipate an A-check, merging an A-check within a C-/D-check will increase the number of A-checks in the long-term.

2.3.2. Assumptions

Ref. [7] defined a list of *major conditions* for maintenance event scheduling, based on aircraft maintenance practice. This chapter adopts the first six of these assumptions (**A.1**–**A.6**) and add two more (**A.7**–**A.8**) necessary to define our approach. **A.1**–**A.7** by far are commonly used among airlines. **A.8** is added due to the fact that airlines do not have their flight schedule for future 4–5 years, thus flexible aircraft routing is assumed for long-term maintenance scheduling.

A.1 Minimum time unit of the aircraft maintenance schedule is 1 DY.

A.2 Aircraft ages by DY, daily FH and FC. The daily utilization, as well as the commercial peak seasons, can be estimated per aircraft according to historical data.

A.3 Each A-/C-check ties up only one hangar (slot) for its total duration.

A.4 A-/C-check priority is defined according to the rule of “earliest deadline first”, i.e., aircraft which has earlier A-/C-check deadline is given higher A-/C-check priority, respectively.

A.5 However, when looking at one particular aircraft, C-check has higher priority than A-check.

A.6 A-check can be merged in C-check, which will not affect the C-check duration or existing A-check slots.

A.7 The duration of an A-/C-check per check label can be estimated according to historical data or can be specified by airline.

A.8 There is flexibility in aircraft routing to accommodate the A-/C-check and the geographical location of the hangars does not have to be specified.

The last assumption is based on the fact that AMCS is a type of strategic problem. The aircraft routings are still unknown since they are only defined a couple of weeks before operations. For this same reason, the location of the aircraft at the time of the maintenance checks is unknown. Therefore, the geographical location of the maintenance checks is not considered in this work. Nevertheless, the formulation presented can be easily adapted to incorporate different hangar locations and constraints regarding the allocation of a given fleet to specific hangar locations.

2.3.3. NOMENCLATURE

This subsection section presents the nomenclature that is used for the rest of the chapter.

AMCS Model Parameters:

d_k	Minimum interval between the start dates of two type k checks.
$e^i_{k\text{-DY}}$	Maximum DY tolerance of type k check interval of aircraft i
$e^i_{k\text{-FH}}$	Maximum FH tolerance of type k check interval of aircraft i
$e^i_{k\text{-FC}}$	Maximum FC tolerance of type k check interval of aircraft i
$\text{fc}_{i,t}$	Average daily FC usage for aircraft i at day t
$\text{fh}_{i,t}$	Average daily FH usage for aircraft i at day t
$I^i_{k\text{-DY}}$	Interval of type k check of aircraft i in terms of DY
$I^i_{k\text{FH}}$	Interval of type k check of aircraft i in terms of FH
$I^i_{k\text{-FC}}$	Interval of type k check of aircraft i in terms of FC
P_d	Penalty for having an aircraft on the ground waiting for a maintenance slot
P_a	Penalty for an aircraft using the tolerance
$\overline{R}_k$	Remaining day threshold of type k check

Other Parameters:

h	Hangar indicator
i	Aircraft indicator
k	Maintenance check type indicator, $k \in K$
N	Total number of aircraft
n_k	The number of hangars for type k check
n_{act}	The number of actions on day t
t	Indicator of calendar day
T	Final day in planning horizon
t_0	First day in planning horizon
Δu	Increment of fleet utilization for discretization
γ	Discount factor

Decision Variables and Related Attributes:

$a_{i,t}$	The attributes of aircraft i in the beginning of the day t
A_t	$A_t = \{a_{i,t} \mid i = 1, 2, \ldots, N\}$
$\mathrm{DY}_{i,t}^k$	Total DY of aircraft i in the beginning of day t for type k check
$\mathrm{FC}_{i,t}^k$	Cumulative FC of aircraft i at t since last type k check
$\mathrm{FH}_{i,t}^k$	Cumulative FH of aircraft i at t for type k check
K	Set of maintenance check types, $K = \{\mathrm{A, B, C, D}\}$
$L_i^k\left(y_{i,t}^k\right)$	Estimated elapsed time of next type k check with label $y_{i,t}^k$
$M_{h,t}^k$	Binary variable to indicate if a type k check can be performed in hangar h on the day t
M_t^k	Hangar capacity of type k check, $M_t^k = \sum_h M_{h,t}^k$
s_t	State variable
S_t	The set of workable states, $S_t = \{s_t \mid s_t \text{ workable}\}$
$R_{i,t}^k$	Remaining fly days of aircraft i before the next type k check
$y_{i,t}^k$	Next maintenance label for of type k check of aircraft i on the day t
$z_{i,t}^k$	The end date of type k check of aircraft i
$\delta_{i,t}^k$	Binary variable to indicate if aircraft i is undergoing a type k check on the day t
$\epsilon_{i,t}^{k\text{-DY}}$	Extra DY before day t if previous type k check is deferred
$\epsilon_{i,t}^{k\text{-FH}}$	Extra FH before day t if previous type k check is deferred
$\epsilon_{i,t}^{k\text{-FC}}$	Extra FC before day t if previous type k check is deferred
$\eta_{i,t}^k$	Binary variable to indicate if aircraft i is grounded and waiting for a slot of type k check on the day t
$\theta_{i,t}^k$	Tolerance usage indicator of type k check of aircraft i on the day t
$\chi_{i,t}^k$	Binary variable to indicate if aircraft i starts a type k check on the day t
x_t	Available action on the day t
x_t^*	The optimal action among $\{x_t\}$
$\mathscr{X}^\pi(s_t)$	Scheduling policy function, $x_t = \mathscr{X}^\pi(s_t)$
π	Scheduling policy
Ψ	$\Psi \in \{\mathrm{FH}, \mathrm{FC}\}$
$\Psi_{i,t}^k$	$\Psi_{i,t}^k \in \{\mathrm{FH}_{i,t}^k, \mathrm{FC}_{i,t}^k\}$
$\psi_{i,t}^k$	$\psi_{i,t}^k \in \{\mathrm{fh}_{i,t}^k, \mathrm{fc}_{i,t}^k\}$
$\mathcal{S}^X(s_t, x_t)$	State transition function from s_t to s_{t+1}, $s_{t+1} = \mathcal{S}^X(s_t, x_t)$
$u_{i,t}^k$	Utilization of aircraft i on day t with respect to type k check
$\bar{u}_t^k$	Mean utilization of fleet on calendar day t for type k check
$C_t(s_t, x_t)$	Contribution of choosing action x_t on s_t
$J_{t,\bar{u}_t^{\mathrm{A}},\bar{u}_t^{\mathrm{C}}}(s_t)$	Cumulative contribution on day t when the fleet has mean utilization $\bar{u}_t^{\mathrm{A}}$ and $\bar{u}_t^{\mathrm{C}}$ for A-check and C-check respectively
$J_{t,\bar{u}_t^{\mathrm{A}},\bar{u}_t^{\mathrm{C}}}^{\min}(s_t)$	$J_{t,\bar{u}_t^{\mathrm{A}},\bar{u}_t^{\mathrm{C}}}^{\min}(s_t) = \min\left\{J_{t,\bar{u}_t^{\mathrm{A}},\bar{u}_t^{\mathrm{C}}}(s_t)\right\}$
$V_t(s_t)$	The value of being in a state s_t

2

2.4. Model Formulation

2.4.1. Maintenance Capacity and Operational Constraints

This chapter only considers one single maintenance location with multiple maintenance check hangars, meaning that all aircraft will undergo letter checks in the main hub of airlines. Although the hangar locations can also be easily incorporated in the constraints of model formulation, this chapter does not consider this aspect. The main reason is that the AMCS is a type of strategic problem. There is no long-term aircraft rotation or flight schedule for verification or validation. Therefore, multiple locations of performing letter checks would become redundant and only complicate the formulation.

In the AMCS problem, the maintenance check capacity can either be expressed as person-hour available during a working day or, equivalently, as the maximum parallel maintenance check allowed per working day (defined as “time slots” or just “slots”). The capacity for each check type is not always constant over time. Airlines and maintenance, repair, and overhaul (MRO) service providers usually have operational constraints that influence the maintenance capacity per day. For instance, during commercial peak season (e.g., the period of New Year, Easter, summer, and Christmas), airlines usually prefer to operate with the maximum fleet. Performing heavy maintenance, such as C-/D-checks, will lead to high commercial revenue loss, and it is also common to have reduced or no checks during weekends and public holidays due to higher labor costs. A final example is that some maintenance slots are pre-allocated to third-party aircraft. Maintenance planners of airlines cannot consider those reserved slots in the AMCS for their fleet. This chapter defines the capacity $M_{h,t}^k$ for hangar h:

$$M_{h,t}^k = \begin{cases} 1 & \text{if hangar } h \text{ is available on day } t \text{ for type } k \text{ slots} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

This parameter has to be defined per day per hangar for the entire time horizon, reflecting capacity variations between peak season and off-peak season and between weekends and regular working days, according to the airline policy. The capacity $M_{h,t}^k$ for hangar h can also be set equal to zero if hangar h is reserved for a specific maintenance event, such as replacing landing gears for an aircraft or performing a type k check of a third-party aircraft.

2.4.2. Decision Space

An action x_t of day t is to perform A-checks or C-checks, or do nothing:

$$x_t = \left\{ \left\{ \chi_{i,t}^k \right\}_{i=1}^N \right\}_{k \in K} \tag{2.2}$$

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

$$\chi_{i,t}^k = \begin{cases} 1 & \text{a type } k \text{ check for aircraft } i \text{ is planned to start at time } t \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

2.4.3. Definition of State

A state vector s_t is defined by the set of attributes that influence our decisions and the available maintenance slots of each check type:

$$s_t = \left\{ \left\{ a_{i,t}^k \right\}_{i=1}^N \right\} \tag{2.4}$$

where, each attribute set $a_{i,t}^k$ contains the information of aircraft i on day t, with respect to check type k:

$$a_{i,t}^k = \{ \underbrace{M_t^k,\ z_{i,t}^k,\ \delta_{i,t}^k,\ \eta_{i,t}^k,}_{\text{Type 1 } \left(a_{i,t}^{(1),k}\right)} \underbrace{\mathrm{DY}_{i,t}^k, \mathrm{FH}_{i,t}^k, \mathrm{FC}_{i,t}^k, \epsilon_{i,t}^{k\text{-DY}}, \epsilon_{i,t}^{k\text{-FH}}, \epsilon_{i,t}^{k\text{-FC}}, \theta_{i,t}^k, y_{i,t}^k,}_{\text{Type 2 } \left(a_{i,t}^{(2),k}\right)}$$
$$\underbrace{L_i^k\left(y_{i,t}^k\right), \mathrm{fh}_{i,t}, \mathrm{fc}_{i,t}}_{\text{Type 3 } \left(a_{i,t}^{(3),k}\right)} \} \tag{2.5}$$

These attributes are described in 2.3.3 and discussed in the next subsection. They can be divided into three types, as showed in Table 2.1.

Table 2.1: Different types of attribute within a state s_t.

Type 1 $a_{i,t}^{(1),k}$	Attributes at time t that impact the action x_t and are modified only when a check starts or ends, or when an aircraft is grounded
Type 2 $a_{i,t}^{(2),k}$	Attributes at time t that are updated every time based on their value at time $t-1$
Type 3 $a_{i,t}^{(3),k}$	Attributes at time t that depend on exogenous information

2.4.4. State Transition

The transition between states in subsequent time steps depends on the actions taken. This can be described by a *state transition function* in which the state s_{t+1} is defined as a function of the initial state s_t and the action x_t chosen in state s_t:

$$\begin{cases} x_t = \mathscr{X}^\pi(s_t) \\ s_{t+1} = \mathcal{S}^{\mathscr{X}}(s_t, x_t) \end{cases} \quad \text{for} \quad t = t_0,\ t_0+1,\ \ldots,\ T \tag{2.6}$$

where $\mathscr{X}^\pi(s_t)$ generates actions based on s_t according to hangar capacities at day t, M_t^k ($k \in K$). The state transition function $\mathcal{S}^{\mathscr{X}}(s_t, x_t)$ describes how the state vector is updated and expresses the fact that an action taken at time t influences the future maintenance activities and capacities. A history of such process, including the sequence of actions and evolution of state, can be represented as:

$$\left(s_{t_0},\ x_{t_0},\ s_{t_0+1},\ x_{t_0+1},\ s_{t_0+2},\ \ldots,\ s_{t-1},\ x_{t-1},\ s_t,\ \ldots,\ s_T,\ x_T,\ s_{T+1}\right) \tag{2.7}$$

The main purpose of the state transition is to renew the attributes over the time horizon. The attributes are updated in two phases: pre-decision (Phase 1) and post-decision

(Phase 2). The goal of the pre-decision phase is to update the hangar capacity and aircraft availability for time t before any making maintenance check decision, and this provides the information about how many hangars can be used to perform maintenance checks and which aircraft is available for operation. In Phase 1, only Type 1 attribute ($a^{(1),k}_{i,t}$) is updated and the resulting attributes ($\tilde{a}^{(1),k}_{i,t}$) from the pre-decision update is defined as pre-decision attributes. On the other hand, the goal of the post-decision phase is to update aircraft usage parameters, according to the action x_t that made on the day t. All 3 type attributes will be updated in Phase 2, and this thesis calls the subsequent attributes from Phase 2 update post-decision attributes. Since the attributes of a state are divided into three types (Table 2.1), the transition of each type of attribute is presented separately in the following sub-sections.

UPDATE OF TYPE 1 ATTRIBUTES

Phase 1, which is called the pre-decision phase (Figure 2.1), only updates the Type 1 attributes $a^{(1),k}_{i,t}$. This phase checks if at time t is the end day for an ongoing aircraft check. The results within the pre-decision attributes of type 1 is $\tilde{a}^{(1),k}_{i,t} = \left\{\tilde{M}^k_t,\ \tilde{z}^k_{i,t},\ \tilde{\delta}^k_{i,t},\ \tilde{\eta}^k_{i,t}\right\}$. The pre-decision update is triggered by verifying if the end date ($z^k_{i,t}$) of an ongoing check is equal to $t-1$ (i.e., if $z^k_{i,t} = t-1$), for any aircraft in the fleet:

$$\tilde{z}^k_{i,t} = \begin{cases} 0 & \text{if } z^k_{i,t} = t-1 \\ z^k_{i,t} & \text{otherwise} \end{cases} \tag{2.8}$$

At the same time, it updates $\delta^k_{i,t}$ to $\tilde{\delta}^k_{i,t}$:

$$\tilde{\delta}^k_{i,t} = \begin{cases} 0 & \text{if } z^k_{i,t} = t-1 \\ 1 & \text{otherwise} \end{cases} \tag{2.9}$$

If the end date of a type k check for an aircraft i is larger than the current calendar day t, it means that there is an aircraft check occurring. And the hangar capacity needs to be updated for time t accordingly:

$$\tilde{M}^k_t = \sum_h M^k_{h,t} - \sum_{i=1}^{N} \tilde{\delta}^k_{i,t} \tag{2.10}$$

where $M^k_{h,t}$ is the maintenance capacity per hangar h at time t. The value of $\tilde{\eta}^k_{i,t}$ is initialized using $\eta^k_{i,t}$, namely, $\tilde{\eta}^k_{i,t} = \eta^k_{i,t}$.

In Phase 2, or the post-decision phase (see Figure 2.1), the action x_t is taken into account to update Type 1 attributes. For all aircraft that start type k check on day t ($\chi^k_{i,t} = 1$), the values of $\delta^k_{i,t}$ and $z^k_{i,t}$ need to be updated. The $z^k_{i,t}$ is updated according to:

$$z^k_{i,t+1} = \begin{cases} t + L^k_i\left(y^k_{i,t}\right) & \text{if } \chi^k_{i,t} = 1 \\ \tilde{z}^k_{i,t} & \text{otherwise} \end{cases} \tag{2.11}$$

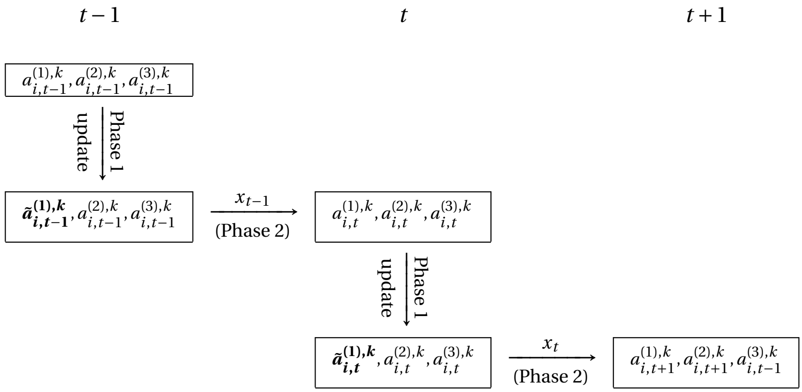


Figure 2.1: A two-phase attribute update mechanism: Phase 1 (Ph-1) updates the set of pre-decision Type 1 attribute $a^{(1),k}_{i,t}$ to $\tilde{a}^{(1),k}_{i,t}$ before making any action; after an action x_t is made, Phase 2 (Ph-2) updates $\tilde{a}^{(1),k}_{i,t}$, $a^{(2),k}_{i,t}$ and $a^{(3),k}_{i,t}$ to $a^{(1),k}_{i,t+1}$, $a^{(2),k}_{i,t+1}$ and $a^{(3),k}_{i,t+1}$.

where $L^k_i\left(y^k_{i,t}\right)$ is the elapse time for maintenance type k, with label $y^k_{i,t}$. Following this update, the values of $\delta^k_{i,t}$ can also be renewed:

$$\delta^k_{i,t+1} = \begin{cases} 0 & \text{if } \chi^k_{i,t} = 1 \\ \tilde{\delta}^k_{i,t} & \text{otherwise} \end{cases} \tag{2.12}$$

Still in the post-decision stage, in some special cases, aircraft reach their inspection intervals, and no maintenance check capacity is available. In these undesirable situations, the aircraft needs to be grounded and put out of operations, waiting for the next maintenance opportunity, and this happens if the usages parameters for time $t+1$ of any aircraft is larger than the respective inspection interval. It first computes the expected usage parameters of $t+1$ as follows:

$$\Delta\mathrm{DY}^k_{i,t+1} = \underbrace{\left(\mathrm{DY}^k_{i,t}+1\right)}_{\mathrm{DY}^k_{i,t+1}} - \underbrace{\left[I^i_{k\text{-DY}} + \left(1-\theta^k_{i,t}\right)e^i_{k\text{-DY}} - \epsilon^{k\text{-DY}}_{i,t}\right]}_{\text{Actual DY Interval of Type } k \text{ Check}} \tag{2.13}$$

$$\Delta\Psi^k_{i,t+1} = \underbrace{\left(\Psi^k_{i,t}+\psi_{i,t}\right)}_{\text{usage parameters of } t+1} - \underbrace{\left[I^i_{k\text{-}\Psi} + \left(1-\theta^k_{i,t}\right)e^i_{k\text{-}\Psi} - \epsilon^{k\text{-}\Psi}_{i,t}\right]}_{\text{Actual Interval of } \Psi\ (\Psi\in\{\mathrm{FH},\mathrm{FC}\}) \text{ of Type } k \text{ Check}} \tag{2.14}$$

where $\Psi^k_{i,t+1} \in \left\{\mathrm{FH}^k_{i,t+1}, \mathrm{FC}^k_{i,t+1}\right\}$, $\Psi \in \{\mathrm{FH}, \mathrm{FC}\}$ and $\psi^k_{i,t+1} \in \left\{\mathrm{fh}^k_{i,t+1}, \mathrm{fc}^k_{i,t+1}\right\}$ are used for convenience. This chapter separates DY from other usage parameters because its utilization update is different from FH or FC. $\mathrm{DY}^k_{i,t}$, $\mathrm{FH}^k_{i,t}$ and $\mathrm{FC}^k_{i,t}$ are the cumulative DY, FH and FC since previous type k check till day t; $I^i_{k\text{-DY}}$, $I^i_{k\text{-FH}}$ and $I^i_{k\text{-FC}}$ refer to the standard interval of type k check; $\left(1-\theta^k_{i,t}\right)e^i_{k\text{-DY}}$, $\left(1-\theta^k_{i,t}\right)e^i_{k\text{-FH}}$ and $\left(1-\theta^k_{i,t}\right)e^i_{k\text{-FC}}$ represent the respective tolerance that can be added to the standard interval; and $\epsilon^{k\text{-DY}}_{i,t}$, $\epsilon^{k\text{-FH}}_{i,t}$ and $\epsilon^{k\text{-FC}}_{i,t}$ represent the amount of DY, FH and FC tolerance used in previous type k check.

The interval tolerance allows maintenance operators to fit the maintenance schedule around maintenance capacity and constraints, and commercial operation demands. However, in the case that tolerance is used in one maintenance cycle, the amount of DY, FH, and FC used from the tolerance needs to be deducted from the maximum usage parameter values for the next maintenance check. In this way, it guarantees that the maximum usage parameters are verified in the long term. The inspection interval tolerance should not be included as a planning option. Even so, it is commonly used in practice to accommodate deviations from the initial schedule.

The aircraft is grounded if any of these previous delta values is greater than 0 and no maintenance check is being performed on this aircraft:

$$\eta^k_{i,t+1} = \begin{cases} 1 & \chi^k_{i,t} = 0, \max\left\{\Delta\mathrm{DY}^k_{i,t+1}, \Delta\mathrm{FH}^k_{i,t+1}, \Delta\mathrm{FC}^k_{i,t+1}\right\} > 0 \\ \tilde{\eta}^k_{i,t} & \text{otherwise} \end{cases} \tag{2.15}$$

UPDATE OF TYPE 2 ATTRIBUTES

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

$$\mathrm{DY}^k_{i,t+1} = \left(1 - \delta^k_{i,t}\right)\left(\mathrm{DY}^k_{i,t} + 1\right) \tag{2.16}$$

$$\Psi^k_{i,t+1} = \left(1 - \delta^k_{i,t}\right)\left[\Psi^k_{i,t} + \left(1 - \delta^{k'}_{i,t}\right)\psi_{i,t}\right] \tag{2.17}$$

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 was scheduled in the previous time step (i.e., $\delta^k_{i,t} = 1$). Otherwise, the parameters are either increased by the average daily aging of the aircraft or kept constant, if a maintenance of the type other than k is scheduled (i.e., $\delta^{k'}_{i,t} = 1$).

The update of Type 2 attributes also includes renewing the tolerance usage variables for each maintenance check type k:

$$\epsilon^{k\text{-DY}}_{i,t+1} = \begin{cases} \max\left\{0, \Psi^k_{i,t} - I^i_{k\text{-DY}}\right\} & \text{if } \chi^k_{i,t} = 1 \\ \epsilon^{k\text{-DY}}_{i,t} & \text{otherwise} \end{cases} \tag{2.18}$$

$$\epsilon^{k\text{-}\Psi}_{i,t+1} = \begin{cases} \max\left\{0, \Psi^k_{i,t} - I^i_{k\text{-}\Psi}\right\} & \text{if } \chi^k_{i,t} = 1 \\ \epsilon^{k\text{-}\Psi}_{i,t} & \text{otherwise} \end{cases} \tag{2.19}$$

where $\Psi \in \{\mathrm{FH}, \mathrm{FC}\}$. (2.18) and (2.19) indicate that the status of tolerance usage of a type k check is the same as the day before if there is no type k check allocated on day t. On the contrary, if a type k check is scheduled before all usage parameters reach maximum, then no tolerance is used and $\epsilon^{k\text{-DY}}_{i,t+1}/\epsilon^{k\text{-FH}}_{i,t+1}/\epsilon^{k\text{-FC}}_{i,t+1}$ are set to 0. If an aircraft has to operate over the limit of a type k check, the corresponding $\epsilon_{i,t+1}^{k\text{-DY}}/\epsilon_{i,t+1}^{k\text{-FH}}/\epsilon_{i,t+1}^{k\text{-FC}}$ are updated according to the difference between the cumulative DY/FH/FC and type k check interval. As a result, the tolerance usage indicators will be renewed:

$$\theta_{i,t+1}^{k} = \begin{cases} 1 & \text{if } \max\left\{\epsilon_{i,t+1}^{k\text{-DY}},\ \epsilon_{i,t+1}^{k\text{-FH}},\ \epsilon_{i,t+1}^{k\text{-FC}}\right\} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

After an action is evaluated, 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:

$$y_{i,t+1}^{k} = \begin{cases} y_{i,t}^{k} + 1 & \text{if } \chi_{i,t}^{k} = 1 \\ y_{i,t}^{k} & \text{otherwise} \end{cases} \tag{2.21}$$

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 maintenance and aircraft utilization data of the airline. They refer to:

- $L_i^k\left(y_{i,t+1}^k\right)$ is the elapsed time specified by airline.
- $\text{fh}_{i,t+1}^k$ is estimated according to historical aircraft FH.
- $\text{fc}_{i,t+1}^k$ is estimated according to historical aircraft FC.

2.4.5. CONSTRAINTS FORMULATION

There are two types of constraints in the AMCS optimization: the interval of each maintenance check type and operational constraints. The maintenance operators of airlines usually schedule the maintenance checks before the corresponding usage parameters reach maximums. That not being possible, in practice, the airline can make use of the interval tolerance. The extra DY/FH/FC used from tolerance must be compensated in the next type k check, as mentioned in Subsection 2.4.4, and this can be described as follows, for each maintenance check type k, aircraft i, and time t:

$$\text{DY}_{i,t+1}^{k} \le I_{k\text{-DY}}^{i} + \left(1 - \theta_{i,t}^{k}\right) e_{k\text{-DY}}^{i} - \epsilon_{i,t}^{k\text{-DY}} \tag{2.22}$$

$$\Psi_{i,t+1}^{k} \le I_{k\text{-}\Psi}^{i} + \left(1 - \theta_{i,t}^{k}\right) e_{k\text{-}\Psi}^{i} - \epsilon_{i,t}^{k\text{-}\Psi} \tag{2.23}$$

where $\Psi \in \{\text{FH},\ \text{FC}\}$; the first term of the right-hand side of each inequality refers to the standard check interval, the second terms adds the tolerance interval, and the last term subtracts the tolerance used in the previous check of the same type.

Before instigating a maintenance action, it is necessary to verify whether or not there are sufficient slots for a type k check in a hangar during the entire maintenance elapse time $L_i^k\left(y_{i,t}^k\right)$, for all aircraft and hangars available:

$$\chi_{i,t}^k \leq \frac{\sum_{\tau=t}^{t+L_i^k\left(y_{i,t}^k\right)} M_{h,\tau}^k}{L_i^k\left(y_{i,t}^k\right)}, \quad k \in K, \quad t \in [t_0, T] \tag{2.24}$$

The operational constraints are required to guarantee that the number of maintenance checks performed in parallel per day does not exceed the hangar capacity, namely:

$$\sum_{i=1}^{N} \delta_{i,t}^k \leq \sum_h M_{h,t}^k, \quad k \in \{\text{A, C}\}, \quad t \in [t_0, T] \tag{2.25}$$

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

- If $d_k > 0$, there can be at most 1 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 $[t, t + d_k)$

The requirement of the start date can be translated in the following equations:

$$\sum_{i=1}^{N} \chi_{i,t}^k \leq \begin{cases} 1 & \text{if } d_k > 0 \text{ and } \sum_{i=1}^{N} \chi_{i,\tau}^k = 0, \ \forall \tau \in [t - d_k, t) \\ M_t^k & \text{otherwise} \end{cases} \tag{2.26}$$

Note that this chapter uses a generic indicator h to represent a maintenance check hangar, based on the assumption **A.8**. If one wants to consider multiple locations of perform the aircraft A-/C-check, each hangar h would have to be associated with a location l_h and the decision variable $\delta_{i,t}^k$ will be replaced by $\delta_{i,t}^{l_h,k}$.

2.4.6. OBJECTIVE FUNCTION

When scheduling aircraft maintenance activities, the most common objectives are minimization of costs [12, 16] or minimization of the unused flight hours (FH) [7, 14]. This dissertation considers the second objective. The cost minimization objective was not considered for three main reasons:

- The available maintenance cost data is unreliable and hard to associate to 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.

Therefore, minimizing the unused FH, and consequently, in the long term, reducing the number of aircraft checks and days on the ground, is considered to be the best objective for the AMCS problem. For an aircraft i, the value of unused FH in a day t is equal to the summation of the FH loss due to a maintenance check scheduled for that day:

$$\sum_{k\in K} \chi_{i,t}^{k}\left(I_{k\text{-FH}}^{i} - \mathrm{FH}_{i,t}^{k}\right) \tag{2.27}$$

The *contribution function* of FH loss on day t is calculated by:

$$C_t(s_t, x_t) = \sum_{k\in K}\sum_{i=1}^{N}\left[\chi_{i,t}^{k}\left(I_{k\text{-FH}}^{i} - \mathrm{FH}_{i,t}^{k}\right) + \left(1-\chi_{i,t}^{k}\right)P_a\theta_{i,t}^{k} + P_d\eta_{i,t}^{k}\right] \tag{2.28}$$

where the first term on the right-hand side reflects the unused FH of aircraft i, the second term is a penalty for aircraft i using an interval tolerance, and the third term is a penalty for having an aircraft on the ground without doing maintenance.

The penalty P_a is introduced due to the fact that the use of tolerance needs to be communicated and approved by the local civil aviation authorities. Therefore, tolerance should not be considered at a scheduling stage or, if inevitable, it should be used as little as possible. The second penalty is introduced to reflect the cost of having an aircraft on the ground and waiting for a maintenance slot since this results in very high costs. It should always be avoided unless it proves to be unfeasible otherwise. For that reason, the value of P_d should always be of a very large magnitude. Our objective is then to minimize the sum of the total contributions for all states visited during the time horizon, discounted by a factor γ. That is, it searches for the optimal AMCS policy (π) that minimizes the contribution of our scheduling decisions over the time horizon $T - t_0$:

$$\min_{\pi} \mathbb{E}\left\{\sum_{t=t_0}^{T}\gamma^{t-t_0}C_t\left(s_t, \mathscr{X}^{\pi}(s_t)\right)\right\} \tag{2.29}$$

where $\mathscr{X}^{\pi}(s_t)$ is the optimal scheduling policy function.

2.4.7. OPTIMIZATION MODEL

After the introduction of state transition, constraints, and objective function, the optimization problem can be described by the following:

$$\min_{\pi} \mathbb{E}\left\{\sum_{t=t_0}^{T}\gamma^{t-t_0}C_t\left(s_t, \mathscr{X}^{\pi}(s_t)\right)\right\} \tag{2.30}$$

subject to:

Constraints (2.11) – (2.26)

The optimal scheduling policy over the time horizon T can be found by recursively computing the Bellman's equation:

$$V_t(s_t) = \min_{x_t}\left\{C_t(s_t, x_t) + \gamma\sum_{s_{t+1}} p\left(s_{t+1}\middle| s_t, x_t\right)V_{t+1}(s_{t+1})\right\} \tag{2.31}$$

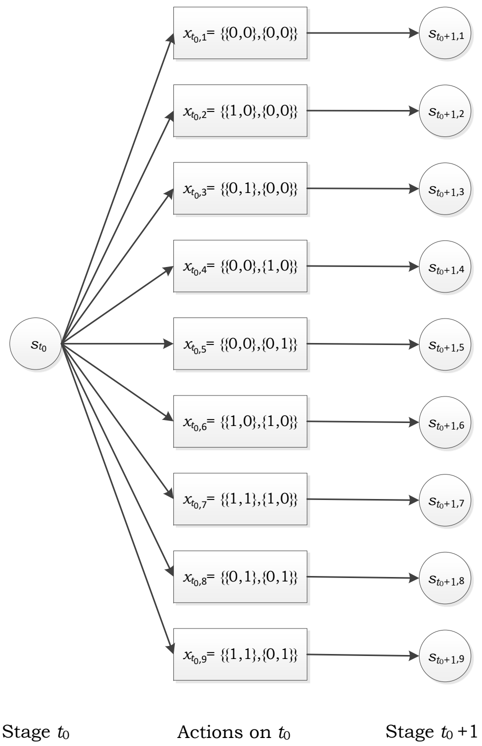


Figure 2.2: An example of state transition from stage t_0 to stage t_0+1 in deterministic AMCS. In this example, only two check types are considered. $\{x_{t_0,i}\}$ is the set of possible actions and $\{s_{t_0+1,i}\}$ is the set of resulting states. In deterministic case, the transition probability is 1, thus there is only one resulting state per action. Similarly from stage t_0+1 on-wards, each $s_{t_0+1,i}$ has the same number of actions as s_{t_0}.

where $s_{t+1} = \mathcal{S}^X(s_t, x_t) = \mathcal{S}^X(s_t, \mathscr{X}^\pi(s_t))$ and $p(s_{t+1}|s_t)$ is the probability of transitioning from state s_t to state s_{t+1}. The Bellman's equation expresses the value of being at each state S_t.

2.5. METHODOLOGY

The AMCS problem has a structure that follows the Markov Decision Process (MDP). Like any other MDP, it can be solved using dynamic programming (DP). The AMCS problem can be divided into stages, each stage referring to one calendar day (indexed by t). For each stage, all possible actions x_t from a state s_t need to be evaluated, and the optimal one x_t^* can be eventually identified. For illustration purpose, this chapter uses Figure 2.2 to depict an example of state transition from stage t_0 to stage t_0+1 (deterministic). In this case, s_{t_0} is the initial state. There are two aircraft, 1 A-check slot and 1 C-check slot on stage t_0. The action vector x_t has the following structure:

$$x_t = \left\{ \underbrace{\{0,0\}}_{\text{A-Check}}, \underbrace{\{0,0\}}_{\text{C-Check}} \right\} \tag{2.32}$$

For each aircraft in (2.32), the first number indicates the action of A-check, and the second number is for C-check. If an A-/C-check starts, the corresponding number is 1, and 0 otherwise, and this gives nine possible actions on stage t_0:

1. no A-check or C-check: $x_{t_0,1} = \{\{0,0\},\{0,0\}\}$
2. A-check on aircraft 1 but no C-check: $x_{t_0,2} = \{\{1,0\},\{0,0\}\}$
3. A-check on aircraft 2 but no C-check: $x_{t_0,3} = \{\{0,1\},\{0,0\}\}$
4. no A-check but C-check on aircraft 1: $x_{t_0,4} = \{\{0,0\},\{1,0\}\}$
5. no A-check but C-check on aircraft 2: $x_{t_0,5} = \{\{0,0\},\{0,1\}\}$
6. merge A- into C-check for aircraft 1 but no A-check on aircraft 2: $x_{t_0,6} = \{\{1,0\},\{1,0\}\}$
7. merge A- into C-check for aircraft 1 and A-check on aircraft 2: $x_{t_0,7} = \{\{1,1\},\{1,0\}\}$
8. no A-check on aircraft 1 but merge A- into C-check for aircraft 2: $x_{t_0,8} = \{\{0,1\},\{0,1\}\}$
9. A-check on aircraft 1 and merge A- into C-check for aircraft 2: $x_{t_0,9} = \{\{1,1\},\{1,0\}\}$

It can be observed that nine possible actions lead to 9 states on stage t_0+1, even for an example of 1 A-check slot and 1 C-check slot (here it assumed that an A-check could be merged into a C-check, in the deterministic case, there is only one resulting state per action). This process repeats as the state transition proceeds, i.e., each state $s_{t_0+1,i}$ has nine different actions, and therefore, will have nine outcome states. As the stage moves forward in time, the number of states within a stage will grow exponentially.

The value associated with each action can be computed using Bellman's optimality equations (2.31). However, solving these for all possible actions is not trivial due to three challenges: the size of the multi-dimensional action vector x_t, the length of the multi-dimensional state vector s_t, and the very large outcome space. These challenges are well known as the "curse of dimensionality" [31]. It is easy to understand these challenges when analyzing the AMCS problem, as the state vector s_t is a tuple that contains the states of N aircraft, and each aircraft has 28 attributes for one type of maintenance check. If one wants to use discretization for each attribute, e.g., to l levels, the total number of levels to access will be $l^{28} \times N$, just for one stage, requiring a large amount of computer memory and also makes it difficult to trace decisions backward. In terms of actions, for each time stage t and capacities M_t^k, there are:

$$\prod_{k\in K} \sum_{m_k=0}^{M_t^k} \frac{N!}{(N-m_k)! \times m_k!} \tag{2.33}$$

possible actions and, if no optimal final state is given, there will be:

$$\prod_{t=t_0}^{T} \prod_{k\in K} \sum_{m_k=0}^{M_t^k} \frac{N!}{(N-m_k)! \times m_k!} \tag{2.34}$$

possible outcomes for the last stage. It means that even for a small case with ten aircraft and one daily slot available for only two check types, there would be 121 possible actions on the first day and more than 1.7 million possible actions just after three days.

In classic DP, computing $V_{t+1}(s_{t+1})$ requires $V_{t+2}(s_{t+2})$, and to obtain $V_{t+2}(s_{t+2})$, the $V_{t+3}(s_{t+3})$ has to be computed and so forth, until t reaching the final stage T. The aforementioned solution process is called backward induction that, if the final state is not known, easily becomes intractable even for the small example of 10 aircraft. Hence, this thesis treat the AMCS problem in a different way, adopting a *forward induction* approach which moves from the initial planning stage towards the future.

This section proposes a forward induction DP based methodology to solve the AMCS problem. It begins with a brief introduction to the forward induction concept in Subsection 2.5.1. After that, it describes a priority solution that is proposed in dealing with the multi-dimensional action vector (Subsection 2.5.2). Subsection 2.5.3 presents a *Thrifty Algorithm* for AMCS, which estimates the implications of an action at the current stage on the remaining planning horizon. Subsection 2.5.4 presents the discretization and aggregation approach adapted to implement the algorithm. The last subsection includes an algorithm complexity analysis.

2.5.1. FORWARD INDUCTION

Forward induction is the process of reasoning forward in time, determining a sequence of optimal actions from an initial state until the end of the time horizon. It comes from the observation that the shortest path from an initial node s_{t_0} to an end node s_{T+1} is equal to the shortest path from the end node to the initial node [32]. That is, determining the optimal solution for the forward shortest path problem is the same as finding the optimal solution for the backward shortest path problem, as computed by the backward induction approach. The idea from the forward induction approach is to move forward in time, continuously searching for the shortest path between the initial node s_{t_0} and the current node being tested s_t. This process is repeated until one has determined the best action for every stage in the time horizon.

Although the forward induction approach would solve the problem of not knowing the final state of our problem, due to too many intermediate states between the initial stage and the final stage of the planning horizon, this approach is still inefficient for AMCS in terms of computation time and storage. For this reason, this thesis incorporates forward induction with three additional components:

- *A maintenance check priorities definition solution*
- *A thrifty algorithm for AMCS*
- *A discretization and state aggregation strategy.*

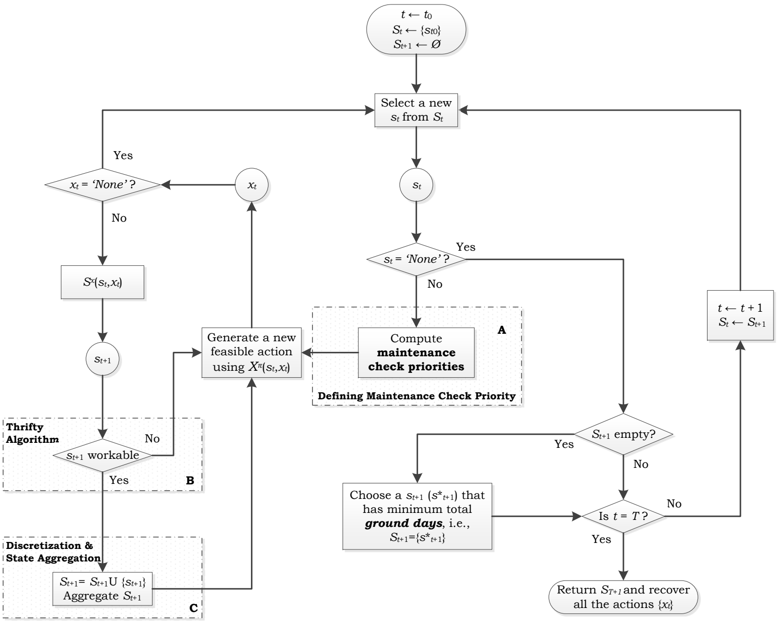


Figure 2.3: Work flow of the DP based methodology. Three main components are incorporated, labeled as blocks **A**, **B** and **C**. The term *ground days* refers to the days when an aircraft is grounded and waiting for a maintenance slot.

These components are labeled as blocks **A**, **B** and **C** respectively, in the work flow diagram of forward induction (Figure 2.3). The first component is used to deal with the multi-dimensional action vector; the second component is used to estimate the consequences of an action on future time steps; the third components is designed to reduce the outcome space to a manageable size. These three components are explained in the following subsections.

2.5.2. DEFINING MAINTENANCE CHECK PRIORITY

Given a state s_t, an action based on s_t is generated:

$$x_t = \mathscr{X}^\pi(s_t) \tag{2.35}$$

Note that x_t means performing m_k type k check (m_k depends on the hangar capacity), which leads to the following combination of aircraft selection for one check action:

$$\prod_{k\in K} \frac{N!}{m_k!(N-m_k)!} \tag{2.36}$$

The number associated outcome states is the same as (2.36).

The number of states can quickly explode due to such a multi-dimensional action vector (an action on multiple items). One solution to this challenge is to assign priorities to each aircraft. In this chapter, it defines maintenance check priorities (**Block A** in

Figure 2.3) according to the rule of "earliest deadline first". However, to know the deadline for a maintenance check, it needs to compute the *remaining utilization* beforehand. Since there are three usage parameters for each check type in the AMCS problem, this gives three different remaining utilization for DY, FH, and FC. A type k check should be scheduled before any of the remaining utilization goes to 0. In this way, the aircraft *remaining utilization* is defined by the fewest days to the next maintenance check:

$$R_{i,t}^{k} = \min\left\{R_{i,t}^{k\text{-DY}}, \quad R_{i,t}^{k\text{-FH}}, \quad R_{i,t}^{k\text{-FC}}\right\} \tag{2.37}$$

The $R_{i,t}^{k\text{-DY}}$, $R_{i,t}^{k\text{-FH}}$ and $R_{i,t}^{k\text{-FC}}$ refer to the remaining operation days with respect to each usage parameter and associated interval specified by the MPD:

$$R_{i,t}^{k\text{-DY}} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{r \le I_{k\text{-DY}}^{i} - \epsilon_{i,t}^{k\text{-DY}} - \mathrm{DY}_{i,t}^{k}\right\} \tag{2.38}$$

$$R_{i,t}^{k\text{-FH}} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{\sum_{\tau=t}^{t+r} \mathrm{fh}_{i,\tau} \le I_{k\text{-FH}}^{i} - \epsilon_{i,t}^{k\text{-FH}} - \mathrm{FH}_{i,t}^{k}\right\} \tag{2.39}$$

$$R_{i,t}^{k\text{-FC}} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{\sum_{\tau=t}^{t+r} \mathrm{fc}_{i,\tau} \le I_{k\text{-FC}}^{i} - \epsilon_{i,t}^{k\text{-FC}} - \mathrm{FC}_{i,t}^{k}\right\} \tag{2.40}$$

where $\mathbb{N}$ is the set of natural numbers for $k \in K$. At any given time t, the remaining utilizations are sorted in ascending order:

$$\tilde{R}_{1,t}^{k}, \tilde{R}_{2,t}^{k}, \tilde{R}_{3,t}^{k}, \ldots, \tilde{R}_{N,t}^{k} \qquad \tilde{R}_{i,t}^{k} \le \tilde{R}_{i+1,t}^{k}, \tilde{R}_{i,t}^{k} \in \left\{R_{i,t}^{k}\right\}_{i=1}^{N} \tag{2.41}$$

The aircraft are sent to maintenance check according to this sorted list, while aircraft with a lower remaining utilization is given a higher check priority. Since the C-/D-check is more restrictive and demanding in terms of resources, it has a higher priority than an A-/B-check. In addition to the available slots and maintenance elapsed time of the check type, the following rules are set for making maintenance check decisions:

(i) No type k check should be scheduled if there is no available hangar for type k check on day t.

(ii) An aircraft i is allocated a type k check only if its remaining utilization is lower than the threshold ($R_{i,t}^{k} \le \overline{R}_{k}$) and there are available slots for type k check.

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

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

After assigning the maintenance check priorities to each aircraft, the combination of aircraft selection for maintenance for one maintenance check action and the number of outcome states is reduced from (2.36) to 1.

2.5.3. THRIFTY ALGORITHM FOR MAINTENANCE CHECK SCHEDULING

Even after reducing the size of outcome space to one action per state, the number of final states n_{act}^{T+1} is tremendous for a large T. It is also necessary to further trim the outcome space, so that forward induction is tractable.

After an action is performed, the DP arrives at s_{t+1}, where $s_{t+1} = \mathcal{S}^{\mathcal{X}}(s_t, x_t)$. Many s_{t+1} states may have the status that some aircraft will have to be grounded to wait for a maintenance check slot in a future stage τ ($\tau > t+1$). Apparently this is what airlines to avoid (unless they have no better option), due to the very high cost of parking aircraft on the ground. Therefore, this thesis proposes to only consider the actions that lead to a *workable* state. It describes *workability* of s_{t+1} by using the following function:

2

$$g(s_{t+1}) = \sum_{k\in K}\sum_{i=1}^{N}\sum_{\tau=t+1}^{T}\eta_{i,\tau}^{k} \tag{2.42}$$

A state s_{t+1}, resulting from being at state s_t and taking action x_t, is said to be *workable* if there exists a sequence of actions x_{t+1}, ..., x_T such that no aircraft has to wait on the ground for an A- or C-check between $t+1$ and T. That is:

$$g(s_{t+1}) = 0 \tag{2.43}$$

The DP-based methodology uses a *Thrifty Algorithm* to check the *workability* of future states (**Block B** in Figure 2.3). That is, for each possible action x_t and resulting state s_{t+1}, it uses an algorithm to check if a sequence of actions exist that guarantee (2.43). If that is the case, s_{t+1} is considered to be *workable*.

The term *thrifty* means allocating maintenance checks to aircraft whenever there is a maintenance opportunity. The *Thrifty Algorithm* serves the purpose of checking the *workability* of a state s_t. For convenience, the rest of the chapter refers to running the *Thrifty Algorithm* to check if (2.43) holds when mentioning "checking *workability*".

2.5.4. DISCRETIZATION AND STATE AGGREGATION

After moving one stage ahead in time for a set of *workable* states s_t, several *workable* s_{t+1} states are produced from a combination of s_t and x_t. Define S_{t+1} to be:

$$S_{t+1} = \left\{ s_{t+1} \middle| s_{t+1} \textit{ workable} \right\} \tag{2.44}$$

Although the *Thrifty Algorithm* can help reduce the outcome space by only keeping the *workable* s_{t+1}, the number of *workable* s_{t+1} is still not bounded, meaning that the number of *workable* states still grows exponentially. This increases the difficulty of saving all *workable* s_{t+1} and tracing the actions backwards, especially After the forward induction move several stages ahead. To prevent the explosion of *workable* states, it needs to restrain the number of *workable* s_{t+1}, from the first stage t_0 to final stage T. That is, giving an upper bound to the number of *workable* s_{t+1} so that it will not increase exponentially along t. For such purpose this thesis resorts to *discretization and state aggregation*.

Discretization is the process of transferring continuous models or variables into discrete counterparts. *State aggregation* refers to clustering the states that have the same properties into a group. Here it uses "properties" to differentiate state attributes, which are the features that at a fleet level, such as mean utilization of fleet or standard deviation of fleet utilization. This thesis divides the outcome space (a set of *workable* s_{t+1}) into several disjunct space regions, where each space region is characterized by a unique tuple of values of some state properties. For the states clustered in the same space region (having the same tuple of state properties), only one single state will be selected to represent such a space region and considered in forward induction for the next stage.

Such *discretization and state aggregation* provides an upper bound to the outcome space, since the number of *workable* s_{t+1} is determined by the number of space regions. One way of collecting *workable* states is to discretize the AMCS problem according to the mean utilization of the fleet for each check type, and then categorize the *workable* states according to the values of the features:

$$\overline{u}^k_{t+1} = \frac{1}{N}\sum_{i=1}^{N} u^k_{i,t+1} \tag{2.45}$$

where $u^k_{i,t+1}$ is the utilization of aircraft i with respect to type k check. This chapter defines individual utilization $u^k_{i,t}$ as the maximum of the ratios between the current value of the usage parameters and their respective maximum values, according to the MPD:

$$u^k_{i,t+1} = \max\left\{\frac{\mathrm{DY}^k_{i,t+1}}{I^i_{k\text{-DY}} - \epsilon^{k\text{-DY}}_{i,t+1}}, \frac{\mathrm{FH}^k_{i,t+1}}{I^i_{k\text{-FH}} - \epsilon^{k\text{-FH}}_{i,t+1}}, \frac{\mathrm{FC}^k_{i,t+1}}{I^i_{k\text{-FC}} - \epsilon^{k\text{-FC}}_{i,t+1}}\right\} \tag{2.46}$$

for $k \in K$. For each check type k, this thesis also gives upper bound $U^k_{\max}$ and lower bound $U^k_{\min}$ to restrict the outcome space region to be discretized, and this significantly improves algorithm efficiency and reduces required computer memory when optimizing AMCS for a large fleet. For instance, given a fleet 1000 aircraft, performing a maintenance check on an aircraft will only impact the overall fleet utilization slightly. In such case, $U^k_{\max}$ and $U^k_{\min}$ can be chosen close to $\bar{u}^k_{t_0}$.

Since tolerance is not allowed in planning unless no feasible schedule can be found, the mean utilization of a fleet ranges typically between 0 and 1 ($U^k_{\max} = 1$ and $U^k_{\min} = 0$), a discretization increment $\Delta u = 0.1$ yields 11^4 space regions in 1 stage, while an increment of $\Delta u = 0.01$ increases the number of space regions in 1 stage increase to 101^4.

Using $\overline{u}^k_{t+1}$ to categorize the set of *workable* s_{t+1} enables us to cover the state properties of all check types, with each tuple $\left(\overline{u}^{\mathrm{A}}_{t+1}, \overline{u}^{\mathrm{B}}_{t+1}, \overline{u}^{\mathrm{C}}_{t+1}, \overline{u}^{\mathrm{D}}_{t+1}\right)$ corresponding to one outcome space region. For each workable state s_{t+1}, the algorithm compute the mean fleet utilization $\bar{u}^k_{t+1}$ for each check type, from (2.45) and (2.46). These features will be further rounded according to the number of decimal points chosen from Δu. For example, if $\Delta u = 0.1$, $\bar{u}^{\mathrm{A}}_{t+1} = 0.345$, then $\bar{u}^{\mathrm{A}}_{t+1}$ can be rounded to 0.3. After that, it continues to compute the cumulative contribution from state s_{t_0} to a specific *workable* state s_{t+1}:

$$J_{t+1,\bar{u}^{\mathrm{A}}_{t+1},\bar{u}^{\mathrm{B}}_{t+1},\bar{u}^{\mathrm{C}}_{t+1},\bar{u}^{\mathrm{D}}_{t+1}}(s_{t+1}) = \begin{cases} J_{t,\bar{u}^{\mathrm{A}}_t,\bar{u}^{\mathrm{B}}_t,\bar{u}^{\mathrm{C}}_t,\bar{u}^{\mathrm{D}}_t}(s_t) + C_t(s_t, x_t) & t > t_0 \\ C_{t_0}\left(s_{t_0}, x_{t_0}\right) & t = t_0 \end{cases} \tag{2.47}$$

where $s_{t+1} = \mathcal{S}^{\mathcal{X}}(s_t, x_t)$, $C_\tau(s_\tau, x_\tau)$ refers to (2.28). If a given space region has no state within it, a cumulative contribution value of infinity ∞ is assumed for that space region.

During forward induction, there will be several workable states grouped into the same outcome space region because of identical $\bar{u}^{\mathrm{A}}_{t+1}$, $\bar{u}^{\mathrm{B}}_{t+1}$, $\bar{u}^{\mathrm{C}}_{t+1}$ and $\bar{u}^{\mathrm{D}}_{t+1}$ after rounding. Then, an aggregation procedure is followed: the state with the lowest cumulative contribution is selected as the representative of its outcome space region, while all others are discarded:

$$s^{*}_{t+1,\bar{u}^{\mathrm{A}}_{t+1},\bar{u}^{\mathrm{B}}_{t+1},\bar{u}^{\mathrm{C}}_{t+1},\bar{u}^{\mathrm{D}}_{t+1}} = \operatorname*{argmin}_{s} \left\{ J_{t+1,\bar{u}^{\mathrm{A}}_{t+1},\bar{u}^{\mathrm{B}}_{t+1},\bar{u}^{\mathrm{C}}_{t+1},\bar{u}^{\mathrm{D}}_{t+1}}(s) \right\} \tag{2.48}$$

In the worst case, no s_t has a subsequent workable s_{t+1}, that is, $g(s_{t+1}) > 0$ for all s_t and $x_t = \mathscr{X}^{\pi}(s_t)$, then the DP-based methodology selects only one $\hat{s}_{t+1}$ according to:

$$\hat{s}_{t+1} = \operatorname*{argmin}_{s,\bar{u}^{\mathrm{A}}_{t},\bar{u}^{\mathrm{B}}_{t},\bar{u}^{\mathrm{C}}_{t},\bar{u}^{\mathrm{D}}_{t}} \left\{ J_{t,\bar{u}^{\mathrm{A}}_{t},\bar{u}^{\mathrm{B}}_{t},\bar{u}^{\mathrm{C}}_{t},\bar{u}^{\mathrm{D}}_{t}}(s) \right\} \tag{2.49}$$

$$S_{t+1} = \{\hat{s}_{t+1}\} \tag{2.50}$$

where the right hand side of (2.49) means choosing the state s among all outcome space regions $\left(\overline{u}^{\mathrm{A}}_{t+1}, \overline{u}^{\mathrm{B}}_{t+1}, \overline{u}^{\mathrm{C}}_{t+1}, \overline{u}^{\mathrm{D}}_{t+1}\right)$. The forward induction then continues from $\hat{s}_{t+1}$.

The procedure from (2.47) to (2.48) repeats until it loops all possible pairs of $\{s_t, x_t\}$ (s_t *workable*). Thus far, it completes the *discretization and state aggregation*, and then the forward induction moves one stage ahead from t to $t+1$. The pseudo code of DP based methodology is presented in Algorithm 1.

2.5.5. ALGORITHM COMPLEXITY

From the perspective of algorithm complexity, the total number of states in our DP based methodology is equivalent to the total number of outcomes, given by (2.34):

$$\prod_{t=t_0}^{T} \prod_{k\in K} \sum_{m_k=0}^{M_t^k} \frac{N!}{m_k!\,(N-m_k)!} \tag{2.51}$$

where N is the fleet size, and M_t^k is the maintenance capacity of type k check. For two maintenance check types (A- and C-checks), given a state s_t at time stage t, following an action x_t, the algorithm has to call the state transition function (2.6) at most $T - t_0 - t + 1$ times to check whether (2.43) holds (from t to T). In each stage t, the number of states depends on the discretization resolution Δu:

$$n_{\text{state}} = \left(1 + \frac{1}{\Delta u}\right)^2 \tag{2.52}$$

Since each state can have at most n_{act} actions, this implies the following relation between the stage t and the number of state transition:

Algorithm 1 A dynamic programming based methodology for AMCS optimization.

Step 1: Initialize Δu $(0 < \Delta u < 1)$, $t \leftarrow t_0$, $S_t \leftarrow \{s_{t_0}\}$, $S_{t+1} \leftarrow \emptyset$
Step 2: Discretize the interval $[0,1]$ with Δu: $0,\ \Delta u,\ 2\Delta u,\ \cdots,\ 1$;
Step 3: For each *workable* $s_t \in S_t$:
Step 3.1: compute and sort the remaining utilization:

$$\tilde{R}^k_{1,t},\ \ldots,\ \tilde{R}^k_{N,t} \quad \tilde{R}^k_{i,t} \le \tilde{R}^k_{i+1,t} \quad \tilde{R}^k_{i,t} \in \left\{R^k_{i,t}\right\}$$

Step 3.2: For each action x_t of $s_t \in S_t$:
Step 3.2.1: Compute $C_t\,(s_t, x_t)$;
Step 3.2.2: Compute s_{t+1} using $s_{t+1} = \mathcal{S}^X\,(s_t, x_t)$;
Step 3.2.3: Check whether s_{t+1} is a *workable* state;
Step 3.2.4: Aggregate s_{t+1} according to

$$J^{\min}_{t+1,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}}(s) = \min_{\{\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}\}}\left\{J_{t+1,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}}(s_{t+1})\right\}$$

$$s^*_{t+1,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}} = \operatorname{argmin}_{\{s,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}\}}\left\{J_{t+1,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}}(s)\right\}$$

$$S_{t+1} = S_{t+1} \cup \left\{s^*_{t+1,\bar{u}^{A}_{t+1},\bar{u}^{B}_{t+1},\bar{u}^{C}_{t+1},\bar{u}^{D}_{t+1}}\right\}$$

Step 4: $t \leftarrow t+1$;
Step 5: If $t \le T$, go to **Step 3**;
Else $s^*_{T+1,\bar{u}^{A}_{T+1},\bar{u}^{B}_{T+1},\bar{u}^{C}_{T+1},\bar{u}^{D}_{T+1}} = \operatorname{argmin}_s \left\{J_{T+1,\bar{u}^{A}_{T+1},\bar{u}^{B}_{T+1},\bar{u}^{C}_{T+1},\bar{u}^{D}_{T+1}}(s)\right\}$;

$$x^*_T(s_T) = \arg_{x_T}\left\{s^*_{T+1,\bar{u}^{A}_{T+1},\bar{u}^{B}_{T+1},\bar{u}^{C}_{T+1},\bar{u}^{D}_{T+1}} = \mathcal{S}^X\,(s_T, x_T)\right\};$$

Step 6: Recover $x^*_{T-1},\ x^*_{T-2},\ \ldots,\ x^*_{t_0+1},\ x^*_{t_0}$

Day t_0:	$n_{\text{act}}(1+T-t_0)$
Day t_0+1:	$n_{\text{state}} n_{\text{act}}(1+T-1-t_0)$
Day t_0+2:	$n_{\text{state}} n_{\text{act}}(1+T-2-t_0)$
...	...
Day $T-2$:	$n_{\text{state}} n_{\text{act}}(1+T-T+1)$
Day $T-1$:	$n_{\text{state}} n_{\text{act}}(1+T-T)$
Day T:	$n_{\text{state}} n_{\text{act}}(1+T-T-1)$

After summing up of all state transitions from t_0 to T, it obtains

$$n_{\text{act}}(1+T-t_0)+n_{\text{state}} n_{\text{act}} \sum_{\tau=0}^{T-t_0} \tau = n_{\text{act}}(1+T-t_0)\left[1+\frac{1}{2}\left(1+\frac{1}{\Delta u}\right)^2 (T-t_0)\right] \tag{2.53}$$

Computing (2.53) gives the maximum number of state transitions in forward induction. Since each of the state transitions generates a new state, this means that the total number of states to be visited is equal to the total number of state transitions in forward induction. Moreover, (2.53) also indicates that the total number of states visited during forward induction depends on the maintenance check capacity (n_{act} is determined by capacity of type k check, M_t^k), and the increment of discretization Δu and planning horizon $T-t_0$. Since (2.53) increases quadratically with T, this means that (2.53) can be much smaller than (2.51) for a large T:

$$n_{\text{act}}(1+T-t_0)\left[1+\frac{1}{2}\left(1+\frac{1}{\Delta u}\right)^2 (T-t_0)\right] \ll \prod_{t=t_0}^{T} \prod_{k\in K} \sum_{m_k=0}^{M_t^k} \frac{N!}{m_k!\,(N-m_k)!} \tag{2.54}$$

2.6. CASE STUDY

In this section, the proposed DP based methodology is evaluated using the aircraft maintenance data from a European airline. The airline distributes the tasks within B-check into successive A-checks (no B-check), and merged the D-checks in C-checks and label them as heavy C-checks. Hence, there are only A- and C-checks in the evaluation. Two case studies are presented in this evaluation. The first case uses data from the historical period from 2013 to 2016. This chapter compares the results obtained by the DP-based methodology with the A- and C-check schedule executed by the airline. However, this comparison is somewhat unfair since the airline in the executed schedule had to take aircraft routing into account and potentially deal with unscheduled maintenance events. Therefore, the second case focuses on the period of 2018-2021 and compares the results from the DP based methodology with the maintenance schedule planned by the airline. This case is also used to support a sensitivity analysis on some of the model parameters. The data set supporting the case study is available on Ref. [33].

2.6.1. TEST CASES

The test fleet is the Airbus A320 family (A319, A320 and A321) operated by the airline, consisting of 45 aircraft. These three aircraft types happen to share the same A- and

C-check intervals and tolerances, in terms of same flight hours, flight cycles and calendar days (Table 2.2). The planning horizon is 4 years in both cases. For 2013-2016, this starts from January 1st of 2013 to December 31st of 2016, while 2018-2021 the planning horizon goes from September 25th of 2017 to December 31st of 2021. For both cases the initial data contains the information for aircraft average monthly utilization; initial fleet status, in terms of DY, FH and FC, and utilization of tolerance in previous inspections; maintenance slots available per day; and average elapsed time of the multiple A- and C-checks labels. The average daily utilization of the aircraft is computed per month and per aircraft type, according to the historic flight data from the airline. On average, it is estimated that the A320 family fleet has a daily utilization of 10.5 FH and 4.7 FC per day.

Table 2.2: A- and C-check intervals and tolerance for the Airbus A319, A320 and A321 [30].

	Check Type	**Calendar Days**	**Flight Hours**	**Flight Cycles**
Inspection intervals	A-Check	120	750	750
	C-Check	730	7500	5000
Tolerance	A-Check	12	75	75
	C-Check	60	500	250

2.6.2. MAINTENANCE CONSTRAINTS AND KEY PERFORMANCE INDICATORS

The maintenance schedule needs to follow a set of operational and capacity constraints, namely, for the A-check:

- there is 1 A-check slot per day from Monday to Thursday during IATA winter (from the last Sunday of October to the last Sunday of March);
- during IATA Summer (from the last Sunday of March to last Sunday of October), there is an extra A-check slot on Tuesdays (2 slots on Tuesday);
- from 2018 onwards, there are 2 A-check slots on Tuesdays (all year) and 2 A-check slot on Wednesdays during IATA Summer;
- there are no A-checks on Fridays, weekends, or public holidays;
- an A-check lasts 1 day and can be merged into a C-check without increasing the C-check elapsed time or affecting the existing available A-check slots.

For the C-check:

- there can be a maximum of 3 C-checks ongoing in parallel;
- there are a minimum of 3 days between the start dates of two C-checks, for resource availability reasons (i.e., $d_C = 3$);
- C-check works are interrupted during weekends and public holidays;
- no C-check can be scheduled during the commercial peak seasons (except some extraordinary occasions in which the airline is forced to have additional slots to avoid aircraft waiting on the ground for a C-check).

Since there are at least three days between two start dates of two successive C-checks, there could be at most 1 C-check starting on a day. The maximum of 2 A-checks on a Tuesday and considering the possibility of merging A-checks into C-check, leads to the combination of daily A- and C-check capacities with 12 possible actions.

The commercial peak seasons of the airline are defined to be between June 1$^{\text{st}}$ and September 30$^{\text{th}}$, two weeks before the New Year's and one week after, and the weeks before and after Easter. The days of the year are converted into calender days where, e.g., the New Year's Day is set as day 1 and Christmas is day 359 of the year (or day 360 if it is a leap year).

To discuss the results, this chapter uses a set of key performance indicators (KPIs) for each type of maintenance check. These are the average FH of the fleet, the total number of checks, the total amount DY/FH/FC used as tolerance, and computation time during the planning horizon. For deterministic problems, the transition probability is made $p\left(s_{t+1}\middle|s_t\right) = 1$ in (2.31). $\overline{R}_{\text{A}}$ and $\overline{R}_{\text{C}}$ are set to 21 and 365, meaning that an aircraft can only be scheduled an A-/C-check if the corresponding remaining operation days is lower than 21/365 days. Since no information is given for discount factor γ, this chapter sets $\gamma = 1$ and the penalty of using tolerance P_a and P_d in (2.28) are given to be 10^8, and this avoids using tolerance in forward induction and grounding the aircraft in the situation of no A- or C-check slot.

The airline schedules the aircraft A- and C-checks separately. The aircraft C-checks are scheduled first with a time horizon of 4-years, followed by planning the A-checks for the next year. In both cases, the airline follows a greedy approach to schedule the checks as close as possible to the end of their intervals. The common conflicts resulting from this approach are then manually solved by the maintenance planner, which anticipates the dates of the checks until a feasible plan is obtained. This manual process is a puzzle, hard to solve for the AMCS close to the peak seasons during which no C-checks can be scheduled, and this results in a sub-optimal schedule that takes a couple of days of work to be fully developed from scratch.

Besides, if an aircraft uses tolerance before undergoing a maintenance check, the extra DY/FH/FC used intolerance must be subtracted from the next maintenance check interval. Namely, the interval to its next A-/C-check becomes shorter. For instance, the A320 family has an A-check interval of 750 FH (see Table 2.2), if an aircraft has to fly 770 FH before undergoing an A-check, then the amount of tolerance used is $770 - 750 = 20$ FH, and the next A-check interval will be $750 - 20 = 730$ FH (this rule has already been considered in the problem formulation (2.22)—(2.23)).

2.6.3. OPTIMIZATION RESULTS FOR 2013-2016

The proposed algorithm is first evaluated for the planning horizon of 2013-2016. Table 2.3 shows a comparison of KPIs between the airline schedule and the DP schedule. It is observed that the average FH increases for both A- and C-checks. For the A-check, there is a growth of 10.4 FH on average per aircraft, which equates to approximately an extra day in commercial operation per aircraft per A-check cycle. This increase has an impact on the number of checks needed for four years. There is a reduction of more than 7% for both A- and C-checks, equivalent to 60 fewer A-checks and seven fewer C-checks. From the perspective of the maintenance cost, assuming that airlines spend on average \$70K—

$350K [2] on a C-check and $10K—$15K on an A-check, the results from the proposed DP-based methodology can result in maintenance costs saving of approximate $1.1M—$3.4M for the fleet of A320 family.

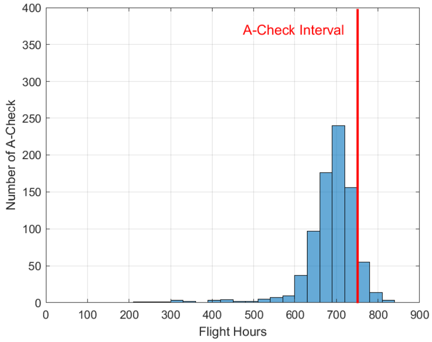
Table 2.3: Descriptive statistics of KPIs for 2013-2016 ($\Delta u = 0.1$ in the DP-based method). For the term "Tolerance Events", if an aircraft uses tolerance (DY/FH/FC) in planning, it is counted as 1 tolerance event.

Type	KPI 2013-2016	Airline	DP-based Method.	Difference
C-Check	Average FH	6795.9	6798.7	0.04%
	Total FH Tolerance	2230	349.2	-84.3%
	Tolerance Events	17	1	-94%
	Extra C-Check Slot	73	0	-100%
	Total C-Check	89	82	-7.9%
A-Check	Average FH	690.8	701.2	1.5%
	Total FH Tolerance	1277	457.4	-64.2%
	Tolerance Events	72	34	-52.8%
	Extra A-Check Slots	101	0	-100%
	Total A-Check	818	758	-7.3%

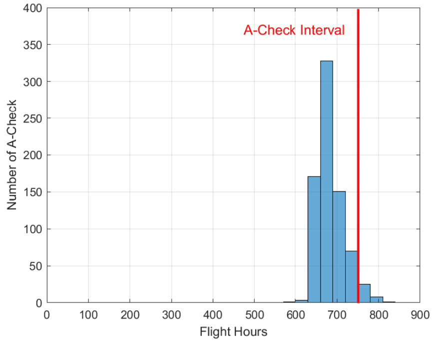
Since it takes 10-30 days to complete a C-check and one day for an A-check, seven reduced C-checks and 60 reduced A-checks are equivalent to approximately 130—270 more days of aircraft availability for revenue generation. One day of operation generates $75K—$120K of revenue and 130—270 more days available for commercial use means an additional $9.8M—$32.4M of revenue for an airline.

The optimized schedule uses tolerances of 349.2 FH and 457.4 for the A- and C-check scheduling, respectively. These are 84% and 64% less than the FH tolerances used by the airline. More importantly, the optimized schedule reduces the frequency of using tolerance (if an aircraft uses tolerance, it is counted as one tolerance events), from 17 to 1 for C-check, and from 72 to 34 for A-check. Recall that using interval tolerance needs to be approved by the national aviation authority, and it is a troublesome process that should not be used recurrently.

It can be observed from Figure 2.4 and 2.5 that the optimized schedule generated by the DP-based methodology concentrates the aircraft FH of A- and C-check close to its corresponding inspection interval. For the A-check, 17% of the checks are scheduled with 95% of the interval used, while for the airline, this value was double, up to 34%. A similar result is obtained for the C-check, where these values are 23% and 43% for the optimized schedule and the airline schedule, respectively. As a result of the greedy approach followed by the airline, the airline has a large number of A- and C-checks scheduled very closed to their interval limit. However, the airline achieved this by using tolerance in 9% of the A-checks and 19% of the C-checks; by scheduling other checks with a quite low interval utilization; and by creating A- and C-check slots, not considered in the optimized schedule, to solve occasional critical situations with several aircraft with high utilization. Note that the optimized results only used tolerance in the checks at the beginning of the time horizon. It was not possible to schedule these checks without using interval tolerance, given the initial state of the fleet and the maintenance slots available.

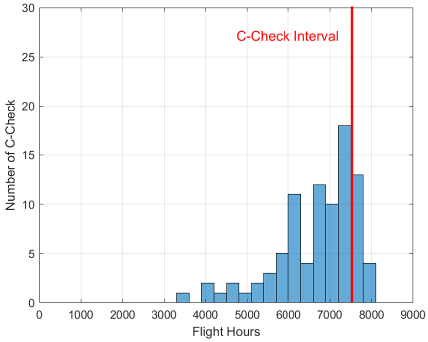


(a) Airline Schedule

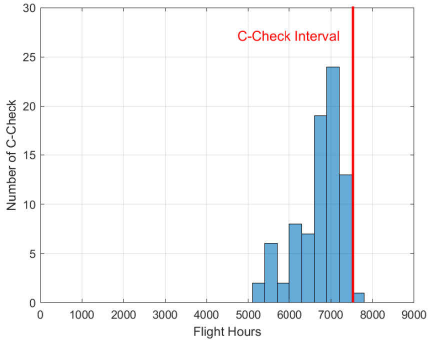


(b) Optimized Schedule

Figure 2.4: Comparison of aircraft FH with respect to A-check between schedule of airline and the optimized schedule.



(a) Airline Schedule



(b) Optimized Schedule

Figure 2.5: Comparison of aircraft FH with respect to C-check between schedule of airline and the optimized schedule.

Table 2.4: Results of A- and C-check scheduling optimization from different discretization resolution, compared with the C-check schedule from Airline.

KPI (2018-2021)	**Airline**	$\Delta u = 1$	$\Delta u = 0.1$	$\Delta u = 0.01$
Objective Value [FH]	—	1.2140×10^5	1.1524×10^5	1.1371×10^5
C-Check Avg. FH	< 6600	6558.1	6615.2	6634.7
Total C-Checks	96	86	88	85
C-Check Tol. DY	> 48	18	18	18
C-Check Tol. FH	> 490	135.3	135.3	135.3
C-Check Tol. FC	0	0	0	0
Tolerance Events	6	4	4	4
A-Check Avg. FH	—	714.3	717.6	714.5
Total A-Checks	895-920*	881	877	881
Tolerance Events	—	0	0	0
Merged A- and C-Check	—	19	18	19
Computation Time [s]	≥ 3 Days	504.9	510.3	20243.5

*Airline Estimation

2.6.4. OPTIMIZATION RESULTS FOR 2018-2021

Although the proposed DP based methodology appears to outperform the planning approach of the airline, based on the KPIs showed in Table 2.3, this comparison is unfair since the airline has to take aircraft routing into account and deal with all kinds of unscheduled maintenance events. To verify and validate the proposed DP based methodology together with the maintenance planners from the airline, this chapter uses it to subsequently generate an optimized A- and C-check schedule for future 2018-2021, and then compare this schedule with the one made by the maintenance planners.

In this test case, both the maintenance planners of the airline and the DP-based methodology plan the 4-year maintenance check schedule using the same input, average aircraft daily utilization, operational constraints, and excluding unscheduled maintenance events and aircraft routing. This subsection compares the KPIs for C-check from both schedules and the optimization results of different discretization resolutions ($\Delta u = 1$, 0.1, and 0.01). Given that the airline only plans the A-check for the coming year, no A-check metrics were compared. The optimized schedules of different discretization resolutions (both A- and C-checks use the same Δu in discretization) are obtained using parallel computing function on a quad-core workstation.

Again it can be seen that the proposed DP based methodology outperforms the planning approach of the airline in terms of KPIs. The optimized schedules reduce the number of C-checks, varying from 8.3% (for $\Delta u = 0.1$) to 11.4% (for $\Delta u = 0.01$), while the same amount of tolerance is used in all three optimized schedules. The tolerance and the number of tolerance events from the optimized schedule are significantly less than what the airline planned. The use of this tolerance is inevitable for aircraft that, at the starting date of the optimization, are already closed to the C-check interval. The number of A-checks vary from 877 (for $\Delta u = 0.1$) and 881 (for $\Delta u = 1$ and $\Delta u = 0.01$), when the airline estimates around 895 to 920 A-checks for these four years. The number of A-merged in the C-checks has little variance among three discretization resolutions.

Besides, an overall trend is found where the level of discretization impacts the so-

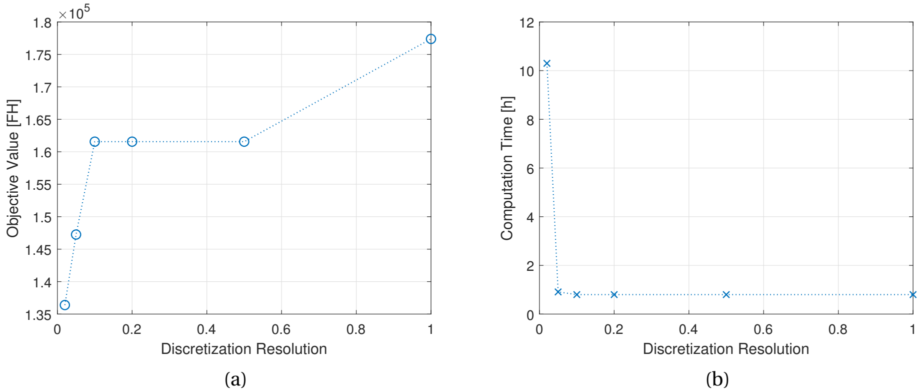


Figure 2.6: (a) correlation between discretization level (resolution) and objective value; (b) correlation between discretization level (resolution) and algorithm computation time.

lution quality and algorithm computation time, as illustrated in Figure 2.6a and Figure 2.6b. In terms of optimality, as expected, the smaller Δu is, the better the results are. However, the KPI's presented in Table 2.4 are no significantly different between the three Δu values tested, indicating that good results can be obtained even with low discretization resolution, and this happens because most *workable* states are in a limited range of the outcome space. A few of the space regions from our discretization do have a *workable* state after aggregation. Nevertheless, there is a trade-off between optimality and computation efficiency. The objective function consistently reduces when the discretization resolution is increased. However, the computational times sharply increase when the discretization resolution is below $\Delta u = 0.1$.

Furthermore, decreasing Δu results in longer computation times. This happens because the number of outcome space region increases with decreasing Δu, e.g., there are 121 outcome space regions when $\Delta u = 0.1$, and 10201 regions for $\Delta u = 0.01$. As a result, the number of representative states of the outcome space region in each stage also increases. In this case study, it is observed that each state transition requires about $0.023s$. For the case $\Delta u = 0.1$, it takes $510.3s$ to obtain an optimized schedule using parallel computing on a quad-core workstation, meaning that the actual computation time should be about $2141.2s$. In particular, during $2141.2s$, there are about $2141.2/0.023 = 9.31 \times 10^4$ state transitions for 1461 stages (from Jan 1$^{\text{st}}$ 2018 to Dec 31$^{\text{st}}$ 2021). The actual computation efforts are still much less than the worst case calculated from (2.53):

$$
\begin{aligned}
\text{comp. time} \quad &= \quad 0.023 \times n_{\text{act}} (1 + T - t_0) \left[1 + \frac{1}{2} \left(1 + \frac{1}{\Delta u} \right)^2 (T - t_0) \right] \\
&= \quad 0.023 \times 7 \times 1461 \times \left[1 + \frac{1}{2} \times 11^2 \times 1460 \right] \approx 2.08 \times 10^7 \ (s) \qquad (2.55)
\end{aligned}
$$

The shorter actual computation time than the worst case is due to the checking of *workability* and *state aggregation*, where it only keeps some *workable* states in each stage, which in the end are sufficient to generate an optimized schedule.

Table 2.5: Sensitivity analysis for having different A- and C-check slots in 2018-2021, the discretization resolution is set to $\Delta u = 0.1$. No A-check tolerance was used in the scenarios tested.

KPI (2018-2021)	Scenario 0	Scenario 1	Scenario 2	Scenario 3
Objective Value [FH]	1.1524×10^5	0.8934×10^5	1.0623×10^5	0.7719×10^5
C-Check Avg. FH	6615.2	6635.0	6699.3	6790.5
Total C-Checks	88	85	85	86
C-Check Tol. DY	18	18	0	0
C-Check Tol. FH	135.3	135.3	23.6	23.6
C-Check Tol. FC	0	0	0	0
A-Check Avg. FH	717.6	738.5	716.5	739.8
Total A-Checks	877	856	890	863
Merge A- and C-check	18	7	22	10
Computation Time [s]	510.3	1780.2	743.0	2625.5

2.6.5. SENSITIVITY ANALYSIS FOR 2018-2021

This subsection investigates the impact of some airline capacity constraints on the results of the AMCS problem, relative to the following four scenarios:

- *Scenario 0*: the baseline scenario, as pre-computed in the previous subsection;
- *Scenario 1*: conditions from Scenario 0 and one additional A-check slot on Friday, weekends, and bank holidays (i.e., one A-check slot every day of the week, plus an extra slot on Tuesdays and Wednesdays during IATA Summer);
- *Scenario 2*: conditions from Scenario 0 and three additional C-checks on weekends and bank holidays (i.e., three C-check slots every day of the week during off-peak seasons, reducing the elapsed time of the C-checks);
- *Scenario 3*: conditions from Scenario 0, Scenario 1, and Scenario 2 combined.

The results, shown in Table 2.5, indicate a natural improvement of the average aircraft utilization and total maintenance checks when increasing either the A-check or the C-check slots. By increasing the number of maintenance slots, it also increases the maintenance check opportunities, given more flexibility in planning the maintenance check closer to their due date. For example, compared with *Scenario 0*, the objective value in *Scenario 3* is reduced by 33%, and the average FH of aircraft is increased by 2.7% and 3.1% for C-check and A-checks, respectively. Consequently, there are two fewer C-checks and 14 fewer A-checks scheduled for *Scenario 3* when compared with *Scenario 0*. For the scenarios involving more C-check slots, it is observed that the results are improved by the fact that more maintenance opportunities exist to merge A-checks and C-checks. In fact, *Scenarios 2* and *3* have around 500 to 600 fewer days on the ground than *Scenarios 1* and *2* respectively. These are days when the aircraft can be used in operation to generate revenue. Furthermore, it is interesting to notice the interdependence between A- and C-checks when analyzing the results from *Scenario 1*. Although only additional A-check slots are added, the results for the C-checks also improve since more A-check slots create more A-check maintenance opportunities. Adding A-check slots gives more flexibility to schedule some of the C-checks that in *Scenario 0* were anticipated to enable the merge with an A-check.

The consideration of extra aircraft maintenance capacity has to be analyzed by the airline by comparing the additional costs of these extra slots and the benefits of having fewer maintenance checks and higher aircraft availability. This analysis is outside the scope of this thesis, but these results are crucial to the airline in assessing such capacity increases (or reductions) scenarios.

2.7. CONCLUSION

A practical dynamic programming (DP) based methodology for the long-term aircraft maintenance check scheduling (AMCS) problem is presented. It integrates all check types, including the operational constraints and maintenance capacity for specific days. The goal was to minimize the total wasted FH interval between checks, hereby increasing aircraft availability in the long run.

The proposed methodology followed a *forward induction* approach, incorporating a maintenance *priority solution* to deal with the multi-dimensional action vector, as well as a *discretization and state aggregation* strategy to reduce outcome space at each time stage. Besides, a *Thrifty Algorithm* was used to estimate the consequence of an action at the current stage on the remaining planning horizon. All these adaptations in the DP framework are novel compared with the classic dynamic programming. The proposed methodology is capable of optimizing AMCS in a matter of minutes for multiple years horizon and heterogeneous aircraft fleets. It is suitable for practical implementation. It can be used not only for scheduling but also, for example, to predict if an airline has sufficient maintenance capacity in the future; or to assess if it is beneficial to expand maintenance capacity with additional hangar slots.

The proposed DP based methodology is evaluated using the case-study of an A320 family fleet from a European airline. Comparing the optimized A- and C-check schedules with the schedule of the airline, it can be inferred that the proposed methodology reduces the total number of A- and C-check, potentially resulting in the long run in maintenance cost savings of about \$1.1M-\$3.4M for a fleet of about 40 aircraft. Besides, the reduction of A- and C-checks implies extra days of aircraft availability for revenue generation. An estimation of \$9.8M-\$32.4M can be generated when the proposed methodology is applied to historical data.

This study is the first to address the AMCS optimization problem despite its relevance for practice, despite its relevance for practice. It opens the door for future research on the topic. For instance, future research can consider the uncertainty associated with both the maintenance check elapsed time and aircraft utilization. These uncertainties will affect not only the schedule robustness but also the computational time needed to find such optimal schedules. One such improvement can be achieved by using approximate dynamic programming, extending the DP-based methodology adopted in this chapter. Another research opportunity is the consideration of the task allocation problem (i.e., the problem of defining the tasks to be performed on each aircraft check) as part of the AMCS problem. Although this would significantly increase the complexity of the problem, it would extend the AMCS problem to good benefits, producing an optimal integrated check and task schedule.

2

REFERENCES

[1] Minister of Justice, *Canadian Aviation Regulations 2012-1, Part I - General Provisions, Subpart 1 - Interpretation,* (2012), (Accessed on September 28, 2017).

[2] S. P. Ackert, *Basics of Aircraft Maintenance Programs for Financiers,* (2010), (Accessed on September 28, 2017).

[3] C. Van Buskirk, B. Dawant, G. Karsai, J. Sprinkle, G. Szokoli, and R. Currer, *Computer-aided aircraft maintenance scheduling*, Tech. Rep. (Institute for Software-Integrated Systems, 2002).

[4] P. Horder, *Airline Operating Costs,* `http://www.dea.univr.it/documenti/Avviso/all/all520253.pdf` (2003), (Accessed on November 15, 2018).

[5] IATA's Maintenance Cost Task Force, *Airline Maintenance Cost Executive Commentary Edition 2019,* (2019), (Accessed on September 11, 2020).

[6] A. Steiner, *A Heuristic Method for Aircraft Maintenance Scheduling under Various Constraints,* in *6th Swiss Transport Research Conference* (Monte Verità, Ascona, 2006).

[7] N. J. Boere, *Air Canada Saves with Aircraft Maintenance Scheduling,* Interfaces **7**, 1 (1977).

[8] M. Etschmaier and P. Franke, *Long-Term Scheduling of Aircraft Overhauls,* in *AGIFORS Symposium* (Broadway, Great Britain, 1969).

[9] H. Bauer-Stämpfli, *Near Optimal Long-Term Scheduling of Aircraft Overhauls by Dynamic Programming,* in *AGIFORS Symposium* (Benalmadena, Spain, 1971).

[10] P. Belobaba, A. Odoni, and C. Barnhart, *Global Airline Industry* (John Wiley and Sons, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2009).

[11] T. A. Feo and J. F. Bard, *Flight Scheduling and Maintenance Base Planning,* Management Science **35**, 1415 (1989).

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

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

[14] M. Başdere and U. Bilge, *Operational aircraft maintenance routing problem with remaining time consideration,* European Journal of Operational Research **235**, 315 (2014).

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

[16] C. Sriram and A. Haghani, *An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment,* Transportation Research Part A **37**, 29 (2003).

[17] C. Senturk, M. S. Kavsaoglu, and M. Nikbay, *Optimization of Aircraft Utilization by Reducing Scheduled Maintenance Downtime,* in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference* (Fort Worth, Texas, 2010).

[18] H. A. Kinnison and T. Siddiqui, *Aviation Maintenance Management, Second Edition,* in *Aviation Maintenance Management, Second Edition* (Mcgraw-Hill Education, 2012).

[19] C. Senturk and I. Ozkol, *The Effects of the Use of Single Task-Oriented Maintenance Concept and More Accurate Letter Check Alternatives on the Reduction of Scheduled Maintenance Downtime of Aircraft,* International Journal of Mechanical Engineering and Robotics Research **7**, 189 (2018).

[20] S. O. Duffuaa and K. S. Al-Sultan, *Mathematical programming approaches for the management of maintenance planning and scheduling,* Journal of Quality in Maintenance Engineering **3**, 163 (1997).

[21] H. Go, J.-S. Kim, and D.-H. Lee, *Operation and Preventive Maintenance Scheduling for Containerships: Mathematical Model and Solution Algorithm,* European Journal of Operational Research **229**, 626 (2013).

[22] A. Kiefera, M. Schildeb, and K. F. Doerner, *Scheduling of Maintenance Work of a Large-Scale Tramway Network,* European Journal of Operational Research **270**, 1158 (2018).

[23] S. Dreyffus, *Richard Bellman on the Birth of Dynamic Programming,* Operations Research **50**, 48 (2002).

[24] E. E. Bomberger, *A Dynamic Programming Approach to a Lot Size Scheduling Problem,* Management Science **12**, 778 (1966).

[25] A. Gascon and R. C. Leachman, *A Dynamic Programming Solution to the Dynamic, Multi-Item Single-Machine Scheduling Problem,* Operations Research **36**, 50 (1988).

[26] E. L. Lawler, *A Dynamic Programming Algorithm for Preemptive Schedule of a Single Machine to Minimize the Number of Late Jobs,* Annals of Operations Research **26**, 125 (1990).

[27] G. H. Graves and C.-Y. Lee, *Scheduling maintenance and semiresumable jobs on a single machine,* Naval Research and Logistics **46**, 845 (1999).

[28] M. Pereira, N. Campodónico, and R. Kelmam, *Long-term Hydro Scheduling based on Stochastic Models,* EPSOM **39**, 1170 (1998).

[29] T. Asamov, D. F. Salas, and W. B. Powell, *SDDP vs. ADP: The Effect of Dimensionality in Multistage Stochastic Optimization for Grid Level Energy Storage*, http://asamov.com/download/SDDP-ADP.pdf (2016), (Accessed on October 1, 2017).

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

[31] W. B. Powell, *Approximate Dynamic Programming - Solving the Curses of Dimensionality* (Wiley-Interscience, New York, 2011).

[32] R. H. Hoppe, *Optimization Theory II - Lecture Notes*, https://www.math.uh.edu/~rohop/Spring_12/index.html (2018), (Accessed on November 15, 2018).

[33] Q. Deng, *Aircraft Maintenance Check Scheduling Data Set*, https://doi.org/10.4121/uuid:1630e6fd-9574-46e8-899e-83037c17bcef (2019), dataset.

3

Optimal Task Allocation for Aircraft Maintenance Check Schedule

In this chapter, the aircraft maintenance planning optimization continues, shifting the focus from long-term deterministic AMCS to task allocation. The task allocation in the aircraft maintenance domain refers to optimally allocating tasks in predefined maintenance checks. It determines the optimal start dates of aircraft maintenance tasks to perform all preventive tasks as close to their due dates as possible, given an optimal letter check schedule. This dissertation is the first to propose a fast constructive heuristic algorithm to optimize the long-term aircraft maintenance task allocation. A case study of a European airline shows that the heuristic algorithm is capable of generating a 4-year task execution plan for a fleet of 45 aircraft in less than 15 minutes, while the optimality gap is within 5% from the solution obtained by a commercial solver.

This chapter is based on the master thesis of M. Witteman [1], supervised by the Q. Deng. The content of this chapter leads to the following paper:

Witteman, M., Deng, Q., and Santos, B. F. (2021). A Bin Packing Approach to Solve the Aircraft Maintenance Task Allocation Problem. *European Journal of Operational Research*. To cite this article, please use the DOI https://doi.org/10.1016/j.ejor.2021.01.027.

3.1. Introduction

Modern airliners have thousands of parts, systems, and components that need to be recurrently maintained after undergoing certain flight hours (FH), flight cycles (FC), calendar days (DY), or months (MO). The FH, FC, DY, and MO are known as usage parameters, and their maximums allowed in operation are defined as inspection intervals. The optimal allocation of the maintenance tasks to the best maintenance opportunities is a challenging problem solved daily by maintenance planners. The most common approach followed is to group tasks into maintenance checks (e.g., A-, B-[1], C- and D-check) to ensure a consistent maintenance program in which all tasks are performed before their associated due dates. A typical A-check includes inspection of the interior or exterior of the airplane with selected areas opened, e.g., checking and servicing the oil, filter replacement, and lubrication [2]. C-check requires thorough inspections of individual systems and components for serviceability and function. D-check[2] uncovers the airframe, supporting structure, and wings for inspection of most structurally significant items.

To determine the optimal start date of the tasks, it is common in practice to adopt a sequential process: first, schedule the A-, C- and D-checks and then allocate maintenance tasks to each check. Although some tasks can quickly be packaged into these letter checks, a large number of other tasks (more than 70% for an Airbus A320 aircraft) are dephased from the intervals of these checks. It means that they either have to be allocated to a more frequent letter check or manually allocated by maintenance operators to different maintenance events based on the suitability of the task to that check and the urgency of performing the task in due time. In practice, both approaches are conducted according to the experience of maintenance planners, leading to inefficiencies.

The task allocation problem (TAP) in aircraft maintenance refers to the process of optimally allocating tasks in predefined maintenance checks. It determines the optimal start dates of aircraft maintenance tasks so that all preventive tasks are performed as close to their due dates as possible. TAP is complicated because of its combinatorial nature, and it has to be solved for the entire fleet at the same time. In real-life applications, multiple aircraft checks can be scheduled in parallel, and tasks allocated to these checks will share the maintenance resources. For example, Figure 3.1 illustrates a case for five C-checks overlapping in time. Maintenance resources include material, equipment, and a set of labor hours from different skills. Furthermore, the allocation process is intricate because the maintenance tasks involved in these checks are usually associated with different intervals and elapsed time.

This chapter proposes a novel approach to efficiently address the TAP, which can quickly solve the problem without compromising the solution quality. Maintenance plans are frequently being affected by flight schedule disruptions or the need for unscheduled maintenance tasks, and they constantly need to be revised or even re-planed [3]. Inspired by the bin packing problem (BPP), this chapter considers pre-scheduled aircraft maintenance checks to be bins of different (time) dimensions and sharing a multi-dimensional capacity, referring to the multiple types of labor skills involved in the execution of the tasks. The items are the tasks that need to be packed in the bins, and they are

[1] B-checks are rarely mentioned in practice. The tasks that could be included in B-checks are commonly incorporated into successive A-checks.

[2] Many airlines merge D-check into C-check and label it as a heavy C-check or structural check.

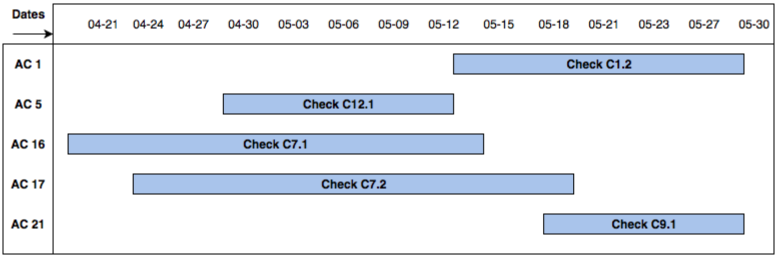


Figure 3.1: Snapshot of maintenance overlap situation between aircraft.

also subject to time constraints that limit the bin options. This chapter formulates the problem as an extension of the variable size bin packing problem (VS-BPP) [4] in which items are repeated within time intervals, and bins have a variable time dimension. This extension of the VS-BPP is named time-constrained VS-BPP (TC-VS-BPP). This formulation is more complicated than the classic BPP and is classified as strongly NP-hard [5]. Therefore, we present a constructive algorithm to solve this problem efficiently. We test this heuristic in a case study using data from a major European airline and compare the results with those obtained using an exact method. The main contribution of this research can be summarized in the following:

- This work is the first to formulate the TAP as a bin packing problem and solve it with an efficient constructive algorithm.
- For the first time, the classic VS-BPP formulation is extended to consider time intervals for allocating repeated items and variable time dimensions for the bins.
- This research adapts the *worst-fit decreasing* algorithm for the classic BPP to efficiently solve the TC-VS-BPP. The resulting constructive algorithm is validated with a real case study and benchmarked against the solution obtained using a commercial linear programming solver.

The outline of this chapter is as follows: Section 3.2 gives an overview of the relevant literature on maintenance related TAPs and bin packing problems (BPPs). The formulation of the TC-VS-BPP for aircraft maintenance is described in Section 3.3. Section 3.4 presents a task allocation framework and an associated heuristic algorithm. Section 3.5 shows a case study from a European airline and the algorithm performance analysis. The last section summarizes the research with concluding remarks and gives an outlook on future work.

3.2. RELATED WORK

This section briefly discusses previous works. It divides the literature overview into two subsections. The first subsection reviews the research works dealing with the TAP for aircraft maintenance, with different perspectives and methodologies. The second subsection discusses the literature on the bin packing problem.

3.2.1. Maintenance Task Allocation

In one of the initial studies on TAP of aircraft maintenance, Ref. [6] combined the aircraft maintenance task allocation with aircraft operation to one single problem. The authors presented a two-stage system that supports maintenance chiefs in planning both aircraft operations and maintenance activities. The first stage assigns the planes to flight operations using a custom-built, multi-level greedy search algorithm. The second stage schedules all maintenance activities according to a constraint satisfaction problem. The authors tested the system with 17 jets, and results indicate that the system can schedule 3750 maintenance activities for a 3-month planning horizon within 20 minutes. The authors also state that the goal was to plan the activities given various constraints: calendar-based actions have to be done within a specific time window; usage-based actions have to be done when the usage clock on a part or subsystem reaches a particular value; personnel has to be available to do the job (mechanics can only do jobs that they are qualified for), and maintenance jobs have to be inspected by a quality/safety inspector and so forth. However, this initial work does not optimize the maintenance schedule given that support for the flight operation was the top priority.

In contrast to Ref. [6], Ref. [3] presented a heuristic for aircraft maintenance planning, aiming at minimizing the overall number of maintenance actions and uniformly distributing the capacity and flying hours over a given time horizon. The main idea was to split the whole process into sub-processes that could be handled computationally fast at the same time. Determining the optimal position of the maintenance actions was the least difficult one, whereas the balancing step was the most challenging one. Even under various settings and constraints, the proposed algorithms have shown to work reliably, fast, and with good optimization results. According to the case study for a 5-year time horizon, the number of tasks scheduled per fleet was around 50–500. The time to compute a new maintenance plan was about 15 minutes.

In practice, many airlines adopt the top-down approach by appropriately grouping maintenance tasks into large packages and fitting them into letter checks. Ref. [7] followed this approach and developed a maintenance item allocation model (MIAM) to cluster aircraft maintenance tasks into packages. The MIAM first simulates the aircraft utilization, calculates when a maintenance item turns due, and then fits each maintenance item into a package. The authors use the concept of *de-escalation* to assess the quality of their MIAM, which can be interpreted as the loss associated with maintenance items being performed more frequently than necessary. The authors proposed a translation of the de-escalation into additional labor costs essential in the long-term to perform extra maintenance activities. According to a case study of a Boeing 737-NG aircraft, the authors claimed that introducing an initial de-escalation, i.e., performing the first base maintenance before its due date, leads to a lower de-escalation labor cost over time. The authors obtained the best result for an initial de-escalation of 30 days, leading to a savings of 248 labor hours (or €13,902) for a single aircraft. The importance of Ref. [7] is that it provides an alternative of assessing maintenance costs using the causal relationship between expense and labor hours.

Maintenance operation costs, in more detail, include the costs of maintenance tools, labor hours, and aircraft spare parts. Each maintenance task associates a cost. Since there are 1000-3000 tasks involved in aircraft maintenance, and many tasks can be performed in parallel, one of the biggest challenges is to execute the right maintenance task at the right time. Assigning priorities to maintenance tasks, such as the rule of "the most urgent task first", can significantly reduce problem complexity. Ref. [8] considered this aspect and presented an optimization method for aircraft maintenance task allocation integrating simulations of aircraft life-cycles. In a real-life application, the authors obtained the best results when sorting the tasks by cost (labor hours) in descending order. In this way, the optimizer allocated the most expensive tasks to maintenance opportunities closer to the end of the lives of the components.

From an efficiency perspective, finding the best maintenance opportunities and allocating maintenance tasks one after another is exceptionally time-consuming. Since each task has some basic properties to indicate similarities, such as ATA code, maintenance interval, zone, and check type, it is convenient to combine several similar tasks into a package and reduce the total number of tasks. Ref. [9] followed this idea and gave different weights on properties to indicate task similarities. Based on engineering experience, weighting factors 0.05, 0.8, 0.05, and 0.1 are assigned to ATA code, maintenance interval, zone, and check, respectively. The authors solved the TAP of an airline using a fuzzy C-means clustering algorithm. Although convergence and improvements were both achieved, the authors stated there are still some pitfalls that need to be investigated, such as the influence of model parameters on solution quality and convergence rate.

In general, the literature on TAP, especially for a long term planning horizon, is very limited. Some of them address TAP on aircraft level [7, 9], while others on fleet level [3, 6, 8]. Even in the research work of TAP in fleet level, the authors tackled task allocation of each aircraft independently, and eventually looped over the entire fleet. Furthermore, none of those related works has assessed the optimality of the proposed models or heuristics. There is no comparison of how close the solution from proposed models or heuristics to the local/global optimum.

3.2.2. THE BIN PACKING PROBLEM

Despite the various task allocation models and methods discussed before, the TAP of aircraft maintenance is very analogous with the bin packing problem (BPP), where for TAP, the maintenance opportunities are equivalent to bins, and maintenance tasks are considered as items. The keys to solving BPP are bin selection and item allocation. For bin selection strategies, Ref. [10] lists four fundamental and widely used algorithms, *next-fit* (NF), *first-fit* (FF), *best-fit* (BF), and *worst-fit* (WF):

- *Next-Fit* (NF): If the item fits in the same bin as the previous item, put it there. Otherwise, open a new bin and put it in there.
- *First-Fit* (FF): Put each item as you come to it into the oldest (earliest opened) bin into which it fits. Only open a new bin if an item does not fit into any previous bin.
- *Best-Fit* (BF): Put items in bins in a way that it maximizes the utilization of the bins that already have been opened.
- *Worst-Fit* (WF): Put each item into the emptiest bin among those with something in them. Only start a new bin if the item does not fit into any bin that has already

been started. If there are two or more bins already started which are tied for emptiest, use the bin opened earliest from among those tied.

If all items are the same size, there is no difference in the four algorithms. Since items are very likely to have different sizes, the allocation of items to bins becomes intricate and time-consuming. And this may involve shifting bin contents continuously until the item list is empty. Thus, some researchers proposed prioritizing the items before putting them in bins. Ref. [11] has suggested some alternatives to the FF and BF. The author states that if the items are sorted in descending order (i.e., the largest item goes first), the worst-case behavior of bin packing problems can be significantly improved. Therefore, it is now a common step to prioritize items before allocation when solving the BPP. The resulting algorithms are the equivalent *first-fit decreasing* (FFD) and *best-fit decreasing* (BFD) algorithms. Similarly, there are also *next-fit decreasing* (NFD) and *worst fit decreasing* (WFD) algorithms.

In practice, not only items can have various sizes, but also bins can have different capacities, and this leads to variable-sized BPP (VS-BPP). VS-BPP is an extension of the classic BPP, in which bins no longer have the same size, and the cost of a bin is proportional to its size [4]. VS-BPP is more challenging since putting items in bins affects the selections of opening new bins later on and item allocations and vice versa. VS-BPP is NP-hard [12]. Researchers tend to solve it using approximation algorithms instead of finding the exact global optimum. [4] listed some algorithms for VS-BPP, such as *next-fit using largest bins only* (NFL), and *first fit decreasing using largest bins and at the end repack to smallest possible bins* (FFDLR). The authors also showed that allowing repacking small bins and shifting bin contents improves algorithm efficiency. And the FFDLR has better worst-case performance than NFL because there is no repacking in the NFL. Ref. [4] further developed a new algorithm *first fit decreasing using the largest bins, but shifting as necessary* (FFDLS) to dynamically shifting bin contents during the construction of packing. Case studies prove that with dynamically shifting bin contents, FFDLS outperforms both NFL and FFDLR in the worst cases.

While Ref. [4] is one of the first works in VS-BPP, research in this topic continues and flourishes in many other studies [12–15]. The main focus of these studies is on the development of algorithms, yet there is no deadline for putting each item in bins. VS-BPP in scheduling, especially maintenance planning, is very distinct from other fields due to time constraints. For example, each maintenance task associates a due date. In VS-BPP, it is equivalent to imposing a deadline for each item (each item has to be put in a bin before a specific time). Besides, and some tasks have to be performed repeatedly. Once the task is executed, we can anticipate the next arrival time of the same task.

The arrival times of items and item allocation deadlines make the aircraft maintenance scheduling related VS-BPP unique and more complex. Some researchers categorize the VS-BPP, in which each item has an associated arrival time and allocation deadline, as time-constrained VS-BPP (TC-VS-BPP). In one of the very few available references, Ref. [16] presents a Markov Chain Monte Carlo (MCMC) heuristic to address the TC-VS-BPP in a working paper. The main difference between TC-VS-BPP and VS-BPP is that in TC-VS-BPP, the arrival times of the items have specific patterns, e.g., a probability distribution in Ref. [16], and each item has to be allocated before a particular deadline. The MCMC heuristic is a combination of local search and Monte Carlo sampling.

It starts with a simple greedy approach to obtain an initial feasible solution. In this step, the authors create two non-ordered lists for bins and items, respectively, and apply the FF algorithm to put items in bins. After that, the authors use MCMC to improve the initial feasible solutions iteratively. One interesting finding from Ref. [16] is that when time constraints are introduced, smaller and faster bins are preferred to meet the deadlines. But in the classical VS-BPP, items are often concentrated in few high capacitated bins. Two main features in TC-VS-BPP, arrival times of the items and deadlines of the items [16], are also common in maintenance scheduling. Since most of the maintenance tasks have deadlines and follow periodic patterns, once a task is performed, we can already anticipate its next execution.

The review of the literature on TAP, BPP, VS-BPP, TC-VS-BPP, and corresponding solution techniques indicates that an aircraft maintenance TAP is similar to TC-VS-BPP in the model formulation in terms of maintenance capacity constraints, availability of each maintenance hangar, the different costs in task execution, workloads of performing tasks, task execution intervals, and deadlines of the maintenance tasks, meaning that the solution strategies, such as NFD/FFD/BFD/WFD, to BPP/VS-BPP/TC-VS-BPP, can be used to address TAP. We propose a constructive heuristic based on the WFD algorithm to solve the long-term aircraft maintenance TAP. The main reason is that more than 55% of the tasks belong to heavy maintenance, and we want to let the available workforce address as many heavy maintenance tasks as possible in aircraft C-/D-checks. In our problem, we are not trying to reduce the number of bins being used — these were already predefined in the maintenance schedule and as a consequence of the overlapping of multiple checks in time. Furthermore, we want to spread the tasks over the multiple bins in such a way that we avoid resource limitations at any point. So the idea is always to allocate the item to the bin with the minimum load (or higher resources available). Since our work focuses on practical application, instead of worst-case performance analysis, we compare the results from the heuristic to a solution from exact methods.

3.3. Problem Formulation

In this section we define the TC-VS-BPP for aircraft maintenance task allcoation. We start the section with specifying the problem and its scope (subsection 3.3.1), followed by a description of the assumptions followed (subsection 3.3.2). In subsection 3.3.3 we introduce some model considerations, including the concept of *time segment* and the generation of the *task items* in our TC-VS-BPP. Finally, in subsection 3.3.4 we present the optimization model formulation.

3.3.1. Basic Concept and Scope

Task Classes

In the aircraft maintenance context, maintenance tasks can represent regular maintenance jobs needed for the continuous airworthiness of the aircraft or repairing works that need to be performed to correct malfunctions or damage. Accordingly, the tasks can be divided into two main classes [17]:

- **Routine Tasks**: these are the regular tasks outlined in a Maintenance Planning Document (MPD) provided by the aircraft manufacture or defined by the airline in their Operator Approved Maintenance Program (OAMP). These tasks have to be scheduled within certain fixed intervals, specified in terms of usage parameters such as FH, FC, and calendar days. A routine task has to be performed before one of the usage parameters reaches the specified interval.

- **Non-Routine Tasks**: these are non-scheduled tasks that can result from defaults or damage identified when executing a routine task, pilot reports, or abnormal events such as hard landings or ground damages. They can also represent abnormal maintenance interventions suggested by, e.g., the aircraft manufacturer (service bulletins) or the regulatory body (airworthiness directives). When generated, these tasks are also associated with a time window for their execution. And this time window can vary from having to perform the task before the next flight to a couple of weeks after they were generated.

Task Intervals

The aircraft maintenance tasks, regardless of being routine or non-routine, have to be allocated to a maintenance event. These events include line maintenance inspections (i.e., performed at the ramp or remote stands during the turn-around time of the aircraft) and hangar inspections. In this article, we only consider the latter and ignore the small tasks usually performed during line maintenance inspections.

Workforce

The available workforce constrains the task allocation to maintenance check; each maintenance task is associated with the workforce requirements to perform the task. The maintenance workforce is divided per skill types (e.g., engines and flight control systems, avionics, aircraft metallic structure, and painting technicians). It is limited per day or shift, according to the daily workforce schedule. In this study, the availability of the workforce per skill is an input to the model. The number of hours needed per skill type is a characteristic of the task, which can only be allocated to a maintenance opportunity if there is enough workforce for all skill types involved in task execution.

Time Horizon

Given that routine tasks have to be scheduled based on intervals and that these intervals are re-started every time the tasks are performed, the TAP should consider a time horizon that is large enough to cover at least two following task executions. The reason being that, otherwise, a possible action could be to delay the first task as much a possible, disregarding the possibility of executing the tasks the next time. And this can result in a poor or unfeasible solution in the long-term. For this reason, given that some tasks having very large intervals (i.e., some are not performed every year), a multi-year planning horizon is adopted.

Sequential Approach

To plan hangar inspection tasks, we follow a sequential approach, consistent with the practice of most airlines, that is, we assume that the aircraft maintenance check scheduling (AMCS) was solved beforehand and that an optimal letter check schedule is provided. According to this schedule, each check is considered as a maintenance opportunity to perform a maintenance task. Consequently, the goal of the TAP is to allocate the maintenance tasks to the opportunities that are as close as possible to their due dates.

3.3.2. ASSUMPTIONS

This research is subject to the following assumptions:

A.1 There are sufficient aircraft spare parts and available maintenance tools, without constraining the optimal allocation of tasks.

A.2 The optimal allocation of tasks is constrained by the workforce available. The optimal distribution of tasks per shift or worker is not considered in the TAP.

A.3 A-check tasks can be performed in a C-check, but not the other way around.

A.4 Non-routine tasks generated while executing other tasks can also be performed during the same check, this is considered by augmenting the task duration and workforce needed according to "non-routine rates" estimated from historical data.

3

The first two assumptions are reasonable, considering that the TAP is a long-term problem and spare parts, maintenance tools and equipment, and workforce are planned following the maintenance schedule. Assumptions **A.3** and **A.4** are common in practice. The first, because the resources, skills, and time needed to perform most heavy maintenance (C-/D-check) tasks are not compatible with the planning of light maintenance checks (A-/B-check). The second, because the differing tasks from a hangar check can result in pressure to perform these tasks another day, eventually causing disruptions in operations. Therefore, airlines usually prefer to pre-allocate a time and workforce buffer in each maintenance check to execute these non-routine tasks.

3.3.3. MODEL CONSIDERATIONS

TIME SEGMENTS

In practice, maintenance operators are typically confronted with situations of overlapped maintenance checks, in which several aircraft undergo the same type of maintenance check at the time and therefore competing for the limited maintenance resources. Figure 3.2 depicts an example of such a schedule, and five aircraft are scheduled to perform C-checks maintenance between Apr 21st and May 30th. During these overlap periods, resources have to be shared, constraining the optimal allocation of tasks.

We divide the planning horizon depicted in Figure 3.2 into time segments. A time segment is created every time the overlap conditions change. In Figure 3.2, the overlap of checks change on Apr 24th and 30th, May 12th, 15th, 18th, 21st and 30th. Therefore, we create seven time segments: Apr 21st–24th, Apr 24th–30th, Apr 30th–May 12th, May 12th–15th, May 15th–18th, May 18th–21st and May 21st–30th. Each time segment of an aircraft is considered to be a bin, with a given duration (in days) and constrained by the labor available on these days for each given skill type. For example, AC-1 has four bins, T4–T8; AC-5 has only one bin, T3; AC-16 has four bins, T1–T4; AC-17 has five bins, T2–T6; AC-21 has two bins, T6 and T7. It is worth mentioning that all the bins and their associated sizes are defined based on the maintenance check schedule and kept open. Unlike the classic BPP, we do not need to open a bin when we allocate items (tasks). For the rest of the chapter, when we refer to TC-VS-BPP, we also imply that all the bins are predetermined.

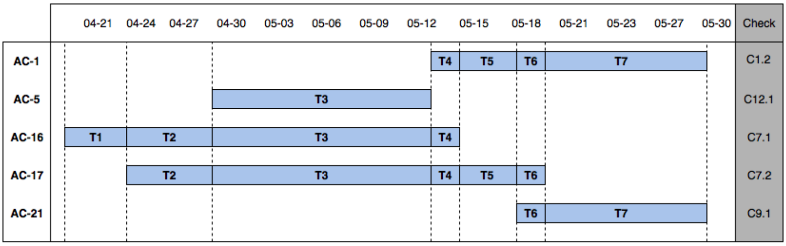


Figure 3.2: Overlapped maintenance checks are divided into several time segments.

Task Items and Maintenance Opportunities

Most routine tasks have to be scheduled more than once for the same aircraft over the time horizon considered. For example, a task that has to be performed in every A-check (about every 7-8 weeks), may have to be executed 38 times in a 5-year horizon. In our approach, we consider each occurrence of these tasks to be an item in our TC-VS-BPP. That is, a routine task that has to be executed at most N times in the planning horizon will be translated into N tasks items in our optimization model. To do so, we have to estimate the maximum number of repetitions in the planning horizon. Table 3.1 illustrates our approach for a given task of a specific aircraft. In this example, the maintenance task has to be performed every ten weeks, while the aircraft A-checks are performed every seven weeks. There are five maintenance events during the time horizon for the execution of the task (four aircraft A-checks and one aircraft C-check, presented in chronological order). This task can be executed from at two times (only in **A2** and **A3**) to five times (in every maintenance check), which can be translated as five task items in the task allocation. The procedure for creating task items and defining the respective maintenance opportunities can be summarized as follows:

- Step 1: The maintenance opportunities for the first execution of the task are determined, according to the state of the task at the start of the planning horizon and its inspection interval.
- Step 2: If the earliest maintenance opportunity for the previous task item is the last maintenance event in the planning horizon, we stop. Otherwise, we create a new task item (next execution).
- Step 3: For the new task item (new execution),
    - Step 3.1: the first maintenance opportunity is the maintenance event right after the earliest maintenance opportunity from the previous task item;
    - Step 3.2: the last maintenance opportunity is the last maintenance event, within the planning horizon, that can be considered before the end of its fixed interval.
    - Step 3.3: all maintenance events between the first and last maintenance opportunities are considered in the set of maintenance opportunities.
- Step 4: Go back to Step 2.

Table 3.1: Illustration of the maintenance opportunities for repeated items of one maintenance task with an inspection interval of 10 weeks (Task 1_1–1_5 represent the 1^{st}–5^{th} execution of the same task). The value of "1" indicates that the associated maintenance check (column) is a possible maintenance opportunity for the execution (row).

Task Execution	**A1 week 1**	**A2 week 8**	**C1 week 12**	**A3 week 15**	**A4 week 22**	**Fictitious opportunity**
Task 1_1	1	1	0	0	0	0
Task 1_2	0	1	1	1	0	0
Task 1_3	0	0	1	1	1	1
Task 1_4	0	0	0	1	1	1
Task 1_5	0	0	0	0	1	1

For the task items which can potentially be allocated to a maintenance check after the end of the planning horizon, we create a *fictitious maintenance opportunity* (bin). The fictitious bin is needed because, eventually, not all task items have to be allocated within the planning horizon to keep the aircraft airworthy. The fictitious bin is added on the day right after the end of the planning horizon, associated with infinite resources and no costs, and it is considered as a bin for all task items that can be scheduled after the end of the planning horizon. This step-wise approach, repeated to all maintenance tasks, will result in a list of task items N_k per aircraft k and the respective set of maintenance opportunities $R_{i,k}$ associated with each task i in the list.

There are several task execution plans for the example presented in Table 3.1. Each plan is associated with a de-escalation cost, depending on the letter checks that the task is executed. We can choose a high-cost plan in which the task is executed in every maintenance check (i.e., five times in the planning horizon) or a low-cost plan in which the task is executed only twice, during the A1 and the A3 checks. Even for two plans with the same number of total executions of a task, the de-escalation is different. For instance, performing the task in **A2**, **C1** and **A3** results in a de-escalation cost of:

$$(10-8)+[10-(12-8)]+[10-(15-12)] = 15 \text{ weeks} \tag{3.1}$$

Executing the task in **A2**, **A3** and **A4** results in a de-escalation cost of:

$$(10-8)+[10-(15-8)]+[10-(22-15)] = 8 \text{ weeks} \tag{3.2}$$

It can be seen from (3.1) and (3.2) that the latter execution plan has a lower de-escalation cost, and the goal of task allocation is to find the task execution plan with the lowest cost, given the resources available and the urgency of other tasks "competing" for the same maintenance opportunities.

3.3.4. PROBLEM FORMULATION

NOMENCLATURE

Sets

i: task indicator

K: set of aircraft

N_k: set of task items for aircraft k ($k \in K$)

T_k: set of time segments for aircraft k ($k \in K$)

$R_{i,k}$: set of time segments for task item i ($i \in N_k$) of aircraft k ($k \in K$)

J: set of skills

$O_{i,k}$ unit set with the task item that follows task item i ($i \in N_k$) of aircraft k ($k \in K$)

Parameters

$c^t_{i,k}$: cost of allocating task item i ($i \in N_k$) from aircraft k ($k \in K$) to maintenance opportunity belonging to time segment t ($t \in T_k$)

$\overline{GR}^j_t$: amount of available labor hours of skill type j ($j \in J$) at time segment t

$GR^j_{i,k}$: amount of labor hours of skill type j prescribed to perform task item i of aircraft k

$\sigma^{j,l}$: "non-routine rate" indicating the amount of labor hours needed from skill type l for every labor-hour prescribed from skill type j (note: $\sigma^{j,j} \geq 1.0 \quad \forall j \in J$)

$\overline{d}_{i,k}$: maximum number of days between rescheduling task item i ($t \in T_k$) for aircraft k ($k \in K$)

d^t: number of days from the start of the planning horizon till maintenance opportunity belonging to time segment t

$\overline{fh}_{i,k}$: maximum number of flight-hours between rescheduling task item i for aircraft k

fh^t: number of accumulated flight-hours from the start of the planning horizon till maintenance opportunity belonging to time segment t

$\overline{fc}_{i,k}$: maximum number of flight-cycles between rescheduling task item i for aircraft k

fc^t: number of accumulated flight-cycles from the start of the planning horizon till maintenance opportunity belonging to time segment t

O_day_i: total days of aircraft operations from the start of the planning horizon to the due date of performing task item i, following the task fix interval and if no resource constraints are considered

$interval_i$: average fix interval for task item i measured in days

$labor\_rate_j$: labor rate, per hour, of skill type j ($j \in J$)

$other\_costs_{i,k}$: non-labor costs associated with task item i ($i \in N_k$) of aircraft k ($k \in K$), such as costs of spare parts and tooling

Decision variables

$x^t_{i,k}$: 1 if task item i is assigned to maintenance opportunity belonging to time segment t for aircraft k, and 0 otherwise

MIXED INTEGER LINEAR PROGRAMMING (MILP) FORMULATION

Given a long-term aircraft maintenance check schedule, this chapter formulates the TAP as a 0-1 MILP model.

$$\min \sum_{k\in K} \sum_{i\in N_k} \sum_{t\in R_{i,k}} c^t_{i,k} \times x^t_{i,k} \tag{3.3}$$

Subject to:

$$\sum_{t\in R_{i,k}} x^t_{i,k} = 1 \qquad \forall i \in N_k \qquad \forall k \in K \tag{3.4}$$

$$\sum_{k\in K} \sum_{i\in N_k} \sum_{j\in J} GR^j_{i,k} \times x^t_{i,k} \times \sigma^{j,l} \leq \overline{GR}^l_t \qquad \forall t \in T_k \qquad \forall l \in J \tag{3.5}$$

$$\sum_{m\in R_{p,k}} d^m \times x^m_{p,k} - \sum_{t\in R_{i,k}} d^t \times x^t_{i,k} \leq \overline{d}_{i,k} \qquad \forall i \in N_k \qquad \forall p \in O_{i,k} \qquad \forall k \in K \tag{3.6}$$

$$\sum_{m\in R_{p,k}} fh^m \times x^m_{p,k} - \sum_{t\in R_{i,k}} fh^t \times x^t_{i,k} \leq \overline{fh}_{i,k} \qquad \forall i \in N_k \qquad \forall p \in O_{i,k} \qquad \forall k \in K \tag{3.7}$$

$$\sum_{m\in R_{p,k}} fc^m \times x^m_{p,k} - \sum_{t\in R_{i,k}} fc^t \times x^t_{i,k} \leq \overline{fc}_{i,k} \qquad \forall i \in N_k \qquad \forall p \in O_{i,k} \qquad \forall k \in K \tag{3.8}$$

$$x^t_{i,k} \in \{0,1\} \qquad \forall k \in K \qquad \forall i \in N_k \qquad \forall t \in T_k \tag{3.9}$$

The objective function (3.3) aims at minimizing the total maintenance costs, which reflect the de-escalation costs associated with scheduling the task earlier than its due date and, consequently, having to perform the task more frequently in the future. To compute these costs, we estimate the due date to allocate the task item beforehand. For example, if a maintenance task is to replace an aircraft component, based on its previous execution date and the associated maintenance interval, we simulate the utilization of the component using the average aircraft's daily utilization. In this way, we can estimate the next due date of replacing this component and its ideal maximum utilization O_day_i. The de-escalation costs can then be calculated by comparing how earlier the task item is allocated when compared with its desired day [8]:

$$c^t_{i,k} = \frac{O\_day_i - d^t}{interval_i} \times \left[\sum_{j\in J} \left(\sum_{l\in J} GR^l_{i,k} \times \sigma^{l,j} \right) \times labor\_rate_j + other\_costs_i \right] \tag{3.10}$$

The de-escalation costs indicated by (3.10) is a reference cost used as a proxy of the goal of scheduling the tasks as later as possible, or as less frequent as possible. In (3.10), the cost of allocating task item i of aircraft k to maintenance opportunity t is a function of the wasted interval of the task (first term), the labor hours required to perform the task (second term), the labor hours cost per labor skill (third term) and additional costs associated with maintenance task i such as the cost for materials or expensive tooling (last term). And this formulation aims at allocating tasks to the maintenance opportunity closer to its due date while giving a higher priority to labor-intensive tasks and tasks involving many labor skills or high additional costs.

Constraints (3.4) guarantee that each task item is allocated exactly once, either to a maintenance event or to the fictitious maintenance event after the planning horizon. Constraints (3.5) make sure that the available labor hours for each skill type is not exceeded in each of the maintenance time segments. The left-hand side of these constraints sums the labor hours needed to perform each task item, including the workforce needed to perform the task and, eventually, associated "non-routine" tasks. These two sets of constraints are the ones that define the classic VS-BPP. The other three set of constraints (3.6)–(3.8) are the features of TC-VS-BPP and also ones that represent the maintenance time-intervals. They imply the arrivals and deadline of tasks. Constraints (3.6) guarantee that a subsequent task item is scheduled within the number of days defined in the fix interval for the respective task, while constraints (3.7) and (3.8) reflect the fix interval in terms of flight-hours and flight-cycles, respectively.

3.4. Task Allocation Framework

The same as BPP, TC-VS-BPP is also NP-hard [18]. Optimal solutions to small TC-VS-BPPs can be obtained using exact methods. Still, unfortunately, when the size of the problem grows, the running times of these exact methods become prohibitive, especially for practical implementations. For this reason, we propose a constructive heuristic to solve the TAP efficiently. The proposed approach is an iterative process based on the WFD algorithm. To the TAP for aircraft maintenance, we start by sorting the tasks from the multiple aircraft into decreasing order of priority and then allocate those tasks one after another to the suitable bin that has a lower load. In this section, we provide details on the proposed constructive heuristic, explaining the general framework, including the input data (Subsection 3.4.1), the necessary pre-computation (Subsection 3.4.2) and the algorithm itself (Subsection 3.4.3).

3.4.1. Input Data

Four sets of input data are needed to formulate and solve the TAP. The first set consists of maintenance task information present in the OAMP for the considered aircraft fleet. This information is not necessarily limited to maintenance tasks described in the MPD. It could include additional maintenance tasks as required by the airline, service bulletins, airworthiness directives, deferred defects, or modifications [17]. Furthermore, information about the last executed date of the routine tasks is used to calculate the first due-date of the maintenance task. The second set includes the estimated daily aircraft utilization, in DY, FH, and FC, of each aircraft for the entire time horizon. For the short term, these values could be obtained using aircraft routes or flight schedules, while in the long run, the most common approach is to use average aircraft utilization per day of the week, per month, or season. It is convenient, however, to use the same input values used to produce the maintenance check schedule. The third set of input is the available workforce per skill type, per day, for the entire time horizon. Again, detailed daily schedules could be provided for the short term, while the average workforce per day can be used for the longer term. The last set of data used is the A- and C-check schedule, defining the starting dates and duration of all checks in the planning horizon for each aircraft in the fleet.

3.4.2. PRE-COMPUTATION

A set of pre-computation steps are necessary before initiating the constructive task allocation algorithm. These steps can be divided into task items and bins related pre-computations. Starting with the task items related steps, maintenance tasks from the same aircraft that have identical intervals, in terms of FH, FC, and DY, are clustered together to reduce the number of tasks to be considered. For the resulting tasks, a set of task items are created, following the procedure explained in Subsection 3.3.3. The following step is to compute the due-dates for the first item of the maintenance tasks. And this is done by considering the initial state of each task (i.e., number of FH, FC, and DY since its previous execution), the task intervals as defined by the OEM or airline, and the simulation of the aircraft utilization over time. Some tasks, such as deferred defects or modifications, can be input already with fixed due dates instead of task intervals.

For the bin related steps, the checks schedule is used to divide the maintenance opportunities into bins, as explained in Subsection 3.3.3. The bins are variable in size and discrete, composed by a set of days. After that, we continue to convert the labor power obtained per day into labor power available per bin.

3.4.3. CONSTRUCTIVE HEURISTIC

A constructive heuristic based on WFD is proposed for task allocation. The pseudo-code of the heuristic is presented in Algorithm 2, while the main procedures of the heuristic are explained next.

SORT TASK LIST

After uploading the input data, the first procedure is to sort the task items list according to the priorities of the items included in the list. The prioritization is done according to a prioritization function $p(i)$ that classifies each task item i. This prioritization function divides task items into three classes:

- *High Priority* – these are items from maintenance tasks that have an interval equal to the interval of the aircraft checks. The allocation process for these items is trivial since those tasks have to be allocated to all equivalent checks in the schedule. This strategy of starting the allocation process with these tasks follows the scheduling practice observed in practice, assigning the workforce necessary to these tasks before starting the allocation of maintenance tasks with more flexibility.

- *Medium Priority* – these are the maintenance tasks dephased from the aircraft checks intervals. Each of these tasks has an interval length larger than the A-check interval (e.g., the task in Table 3.1) and hence they will not necessarily be allocated to every maintenance check.

- *Low Priority* – these are the maintenance tasks with a low frequency of occurrence. They are dephased from the aircraft checks by, at least, being able to skip at least one A-check from any day within the planning horizon. These tasks have some flexibility, and they can be allocated at last.

The tasks within each of these classes are sorted by the maintenance costs, as expressed in the second and third terms in (3.10).

TASK ITEMS LOOP

Task item loop (TIL) is the main procedure of the algorithm. The goal is to choose the best maintenance opportunity that minimizes the maintenance costs, as defined in (3.10), and to select from the bins the one that less compromises the best allocation of subsequent task items. After sorting all the task items according to their costs, the first task item has the highest priority; the second task item has the second-highest priority, and so forth. We define a list of bins that would allow a feasible allocation of task item i before the associate task interval is expired according to the maintenance check schedule. A *fictitious* bin (t_0) is added to this list of bins in case none of the available bins has enough resources to allocate the task item. Other than that, we will not create any new bin during the task allocation.

After that, we sort the available bins for task item i according to the maintenance resources within bins in descending order. Namely, the bin with the most resources is always the first to assign the task in it. After that, the allocation of each task item following a "worst bin" selection process in the fourth step. Therefore, the TIL procedure gives a higher preference to the bins closer to the due date of the task item and, among these bins, to the ones that have more available maintenance resources.

ALLOCATION OF TASKS TO BINS

The next procedure is to allocate the task items to a bin, following the sorted list of bins. If the bin under consideration has enough available labor hours for the necessary skills to perform the respective maintenance task, we allocate an item to the bin. In this case, we subtract the labor hours consumed to execute the task from the total available labor hours from that bin. The next step is to check the need to remove the task that has been allocated. For a routine task, we simulate the evolution of usage parameters after allocation and estimate its new date according to aircraft daily utilization. If the next due date is beyond the end of the time horizon, we just remove the task from the task item list. For a non-routine task, since they are not recurrently performed, we generate a new due date after the end of the planning horizon. For the case of running out of bins to which the task can be allocated, we generate an alert and put the task into fictitious bin t_0. This fictitious bin includes all the tasks that have not been allocated to any available bin, and we will inform the maintenance controller and let them address those tasks.

Algorithm 2 Task Allocation Algorithm

$N \leftarrow$ set of task items from all aircraft, $N = \cup N_k$
$\overline{GR}_t^j \leftarrow$ available labor hours from skill j in bin t
$GR_{i,k}^j \leftarrow$ amount of labor hours of skill j prescribed to perform task item i of aircraft k
$\sigma_{j,m} \leftarrow$ "non-routine rate" from skill m from every hour of skill j

procedure SORT TASK ITEMS LIST
  Sort and reindex N so that $p(i_1) \geq p(i_2) \geq \ldots \geq p(i_n)$ ▷ Prioritization of task items
end procedure

procedure TASK ITEMS LOOP
  while $N \neq \emptyset$ **do**
    Select i from N ▷ Select the first task in the list
    $R_i \leftarrow R_{i,k} \cup t_0$ ▷ Add t_0 as a fictitious opportunity
    Sort and reindex R_i so that $\sum_{j \in J} \overline{GR}_{t_{i,1}}^j \geq \sum_{j \in J} \overline{GR}_{t_{i,2}}^j \geq \ldots \geq \sum_{j \in J} \overline{GR}_{t_{i,n}}^j$
    procedure ALLOCATE TO BIN
      $n \leftarrow 0$
      while $n < |R_i|$ **do**
        $n \leftarrow n+1$
        if $\overline{GR}_t^j \geq \sum_{j \in J} GR_{i,k}^j \times \sigma_{j,m} \quad \forall m \in J$ **then**
          Allocate i to $t_{i,n}$
          Set $\overline{GR}_{t_{i,n}}^j = \overline{GR}_{t_{i,n}}^j - \sum_{j \in J} GR_{i,k}^j \times \sigma_{j,m} \quad \forall m \in J$
          Compute next due-date for task item i
          if Next due-date not within time horizon **then**
            $N \leftarrow N \setminus \{i\}$ ▷ Remove the maintenance task
          else
            Sort Task Items List
          end if
          break
        end if
        if $n = |R_i|$ **then** ▷ In case of no allocation possible
          Allocate i to t_0
          Report Alert
        end if
      end while
    end procedure
  end while
end procedure

3.5. Case Study

In this section, we present a case study on a major European airline and illustrate the applicability of the TAP approach. The input data includes aircraft utilization, a 4-year maintenance schedule generated by the dynamic programming based methodology described in the paper of Ref. [19], task information from a heterogeneous fleet of 45 aircraft, and an associated estimation of available workforce per day. Our airline partner currently follows a manual process to allocate the aircraft maintenance tasks to checks, supported by a digital solution that keeps track of the open tasks and suggests a prioritization of maintenance activities. There are two maintenance planners in the airline doing this job for the entire fleet.

We consider eight skill types and that the productivity factor of each worker is equivalent to 4.8 productive labor hours per day, following the airline practice. The remaining hours of the labor shift are dedicated to transitioning meetings between work shifts, collection of materials or equipment, obtaining information about the maintenance task, reporting, and ancillary activities.

The results from this case study are discussed in subsection 3.5.1, followed by an analysis of the current airline practice of not performing any aircraft C-check tasks in an A-check (subsection 3.5.2). In Subsection 3.5.3, we validate the results obtained using the proposed task allocation heuristic. We suggest assessing the algorithm performance by comparing it with the solution obtained when using an exact method for solving the MILP presented in Section 3.3.4. Furthermore, all results obtained by the proposed heuristic were validated by the maintenance planners of the airline partner.

3.5.1. Optimization Results

We apply the proposed task allocation algorithm to the case study, following the airline current policy of not allowing to allocate C-check tasks to A-check maintenance opportunities. The problem was solved in less than 14 minutes by the algorithm. The outcome is a 4-year, fleet-wide task allocation plan that satisfies labor-hour constraints and tasks fix intervals. The plan includes around 85 thousand task items, from which 24% of them are C-check tasks, and 76% are A-check tasks. Despite this, the C-check tasks consume about 65.5% of the labor hours allocated to perform the tasks. The algorithm achieves an average de-escalation of 205 days for C-check tasks and 19.3 days for A-check tasks.

Figure 3.3 shows the distribution of labor hours per skill for the maintenance of all aircraft in the fleet for the full planning horizon. There is significant diversity in the required labor hours among the aircraft, and the difference in aircraft age, the number of C-check events in the maintenance schedule, and the differences in terms of aircraft utilization cause this diversity. For instance, aircraft AC-41 is phased-out a few days after the start of the planning horizon, while AC-24 is phased out one and half years after the beginning of the planning horizon, following a minor C-check and 10 A-checks. Similarly, it is possible to identify the aircraft that perform a C-check early in the planning horizon and hence have to undergo three C-check before the end of the planning horizon. And this applies to aircraft AC-25, AC-26, and AC-29.

To analyze the maintenance plan in more detail, we decided to focus on the overlap situation presented in Figure 3.2. The allocation of labor hours per time segment is depicted in Figure 3.4. In this figure, there are eight bars per time-segment, representing the eight different skill types. We observe that the first six time-segments consume all the available labor hours of the Group 2 skill type. And this restricts the allocation of tasks requiring labor hours from Group 2 skill for AC-5, AC-16, and AC-17 since these aircraft will have a fully constrained overlap situation. And this forces some of the A-check tasks from these aircraft to be allocated to a previous A-check. Similarly, there are also C-check tasks being anticipated at an earlier C-check. In the latter case, it means that some components are inspected or replaced about two years earlier than intended.

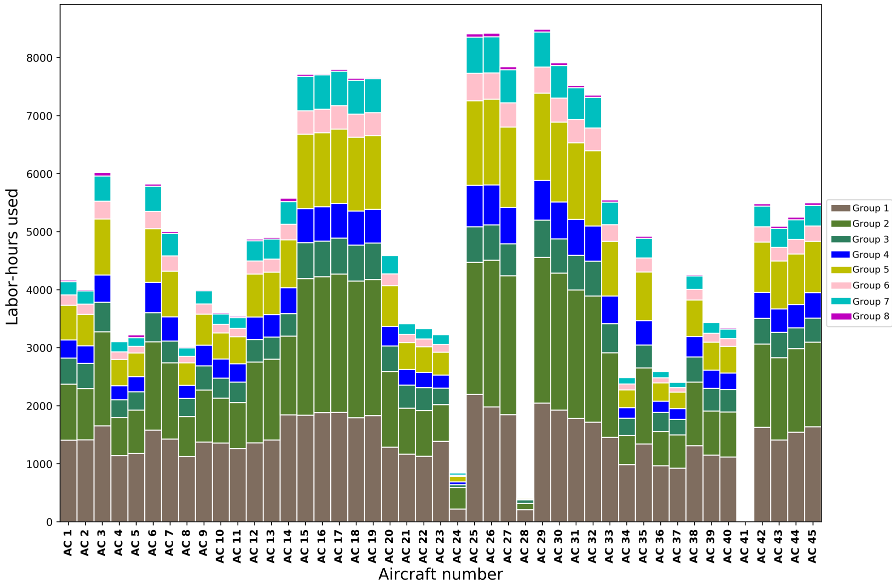


Figure 3.3: Labor hours distribution per aircraft and skill type.

3.5.2. FLEXIBLE TASK ALLOCATION POLICY

In this subsection, we question the current airline policy of not allocating any C-check task to A-check maintenance opportunities, even though we observe that there is a surplus of labor hours in the A-checks scheduled. Several small C-check tasks would fit in an A-check, in terms of time and resources needed. For this reason, we performed a simulation in which these C-check tasks are allowed to be allocated to A-check opportunities. We carry out the analysis considering different thresholds for the size of these tasks. After discussing with maintenance planners from the airline, we agree on using the labor hours needed for the task as the reference metric for task size, and to consider a threshold varying from zero to 2.5 labor hours.

The simulation results (presented in Figure 3.5) indicate that the de-escalation of C-check tasks can be reduced from 205 days to 132 days when allowing C-check tasks within 2.5 labor hours to be executed on A-check opportunities. From the results, it can also be concluded that the marginal gain of extending the threshold reduces as the threshold increases in value. In fact, it can be inferred from Figure 3.5 that, for this air-

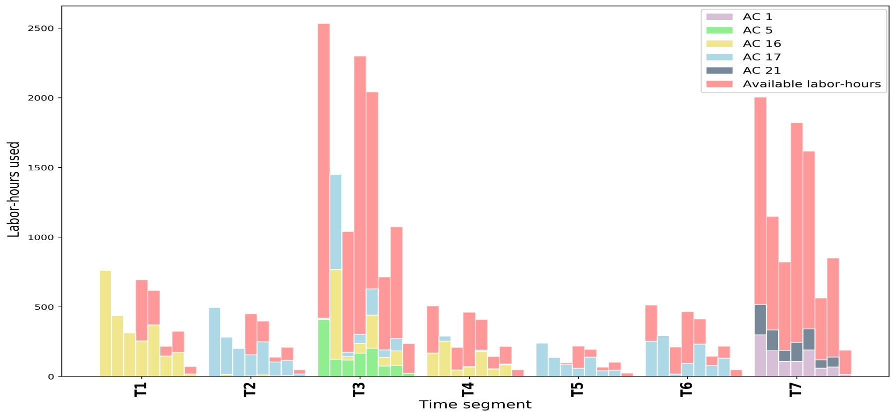


Figure 3.4: The amount of labor hours used for each skill type during the time segments within the overlap situation. Each time segment has eight different bars and each bar represent a particular skill type (Group 1, Group 2,..., Group 8).

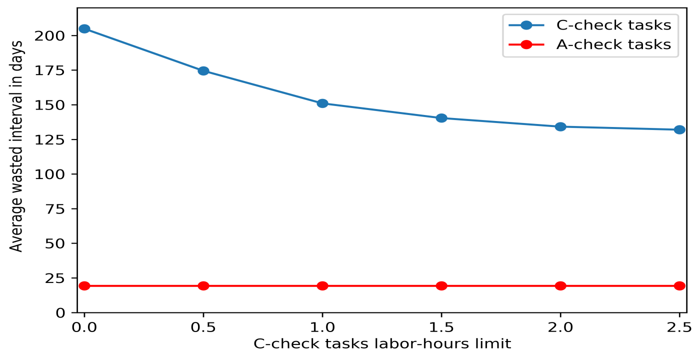


Figure 3.5: Average wasted RUL in days for increased C-check task labor hours thresholds.

line, after a labor hours threshold of 2.0 or 2.5, there are barely any benefits of extending this threshold. The reason being that very few C-check tasks consume more than 2.5 labor hours and can still be allocated in an A-check without compromising the allocation of the A-check tasks to their best A-check opportunities.

3.5.3. Algorithm Performance Analysis

To analyze the performance of the task allocation algorithm, we compare it with the performance of an approach using an exact method to solve the TAP formulated in subsection 3.3.4. To provide a more detailed comparison, we decided to vary the productivity factor of the workforce, from the initial considered 4.8 labor hours per day to a restricted case of 3.2 labor hours per day.

To compute solutions with the exact method in a reasonable time for the more re-

Table 3.2: Simulation results of performance analysis.

Productivity	Computational time (s)		Solution Gap
Labor hours	MILP solver	Heuristic	Heuristic vs. Solver
4.80	1,135	773	0.03%
4.72	1,148	775	0.03%
4.64	1,154	776	0.03%
4.56	1,159	778	0.04%
4.48	1,168	779	0.05%
4.40	1,172	787	0.11%
4.32	1,181	792	0.23%
4.24	1,186	798	0.36%
4.16	1,187	803	0.54%
4.08	1,195	811	0.81%
4.00	1,639	818	1.17%
3.92	1,821	822	1.44%
3.84	1,903	828	1.61%
3.76	2,570	835	1.43%
3.68	3,097	839	1.90%
3.60	3,857	846	2.45%
3.52	4,702	851	2.88%
3.44	5,679	861	3.38%
3.36	8,243	866	3.89%
3.28	13,828	871	4.47%
3.20	17,636	879	4.95%

stricted cases, we follow an iterative process for the creation of task items and maintenance opportunities for each maintenance task (Subsection 3.3.3). That is, we initially run the MILP, then add new items and maintenance opportunities for those task items that had the constraints violated and rerun the MILP until the problem becomes feasible. For the 4.8 labor hours case, the MILP formulation resulted in 1.15 million decision variables and 373 thousand constraints. The task allocation algorithm is coded in Python 3.7, while the exact method is addressed using the commercial solver Gurobi. The results from both approaches are computed on an Intel Core i7 2.6 GHz laptop with 8GB ram.

We summarize the results in Table 3.2. Each line of Table 3.2 compares the computation times and presents the optimality gap for a given productivity factor between two different approaches, where the results obtained from the solver is used as a reference. While the computation time of the exact method (MILP solver) explodes with the decrease of the productivity factor, the same does not happen to the proposed heuristic algorithm. The proposed heuristic is 30% faster than the exact method for the default productivity factor of 4.8 labor hours, and the optimality gap is only 0.03%. Even though the productivity labor hours decrease to 3.2, the optimality gap is still within 5%.

It is worth mentioning that for the most constrained test case, the exact method requires about 4.9 hours to compute the optimal solution, while the proposed heuristic needs less than 15 minutes. In summary, from a perspective of solution quality, the solution gap between the heuristic algorithm and the optimal solution is within 5% for all test cases. For cases where the productivity factor was higher than 4.0 labor hours, the solution gap is below 1%. And this confirms that the heuristic is capable of producing good solutions in minutes for a realistic TAP with a fleet of 45 aircraft.

3.6. Conclusion

The task allocation problem (TAP) of aircraft maintenance is defined as assigning tasks to their optimal maintenance opportunities. In this research, we formulate TAP as a time-constrained variable-size bin packing problem (TC-VS-BPP), in which we treat maintenance opportunities as bins and the tasks as items, and there are time constraints on both bins and items. TC-VS-BPP is NP-hard and, therefore, challenging to solve for large case instances. For this reason, we proposed a constructive heuristic to solve the TAP (TC-VS-BPP). The proposed approach is an efficient iterative process based on the worst-fit decreasing (WFD) algorithm. According to a real-life case study on a heterogeneous fleet of 45 aircraft, the heuristic is more than 30% faster than an exact method, while the solution gap is smaller than 0.1%. For the most restricted test case, the solution from the heuristic is only 5% worse than the solution obtained from the exact method, while being much faster. The computation time of TAP is essential in the aircraft maintenance domain since changes to the priority/urgency of existing tasks or new (non-routine) tasks can require running the proposed constructive heuristic many times per day. Therefore, an algorithm that runs in a reasonable and stable computational time, regardless of how restrictive is the problem, is something very useful.

During the case study, We are told that some airline technicians work just part-time at the hangar, and we overestimated the maintenance capacity if we set the productivity labor hours to 8 (all technicians are working full time, 8 hours a day). The maintenance capacity constraint (3.5) is not the main restriction during the task allocation process. Since there is no other data to support sensitivity analysis, we change the productivity labor hours to test the proposed heuristic in a more constrained context.

The research presented in this chapter is also one of the requirements from the airline, continuing the work of aircraft maintenance check scheduling optimization described in [19]. The methodology in [19] first determines the optimal start dates of all maintenance checks for the entire fleet, and the optimal maintenance check schedule indicates in which checks a maintenance task can be allocated. Otherwise, it is very time-consuming to know when, which aircraft, and what maintenance tasks should be performed without a maintenance check schedule. The maintenance task allocation results are the task execution plans for all maintenance checks, which help the technicians execute the right task, on the right aircraft, at the right time.

We structure the task allocation problem of aircraft maintenance as a bin packing problem (BPP) so that it can be solved quickly using the worst-fit decreasing algorithm. Whenever unscheduled aircraft maintenance tasks occur, we can use the methodology presented in [19] to obtain a new maintenance check schedule, and then apply the task allocation framework to update the tasks accordingly. The task allocation framework is suitable for real-life applications. It can provide near-optimal solutions to the TAP, significantly reducing the workload currently required in practice for the creation of maintenance plans. Besides, given that it runs in minutes, it can potentially be used to dynamically adjust the task allocation plans given flight schedule disruptions during operations or emergency of unscheduled tasks during the execution of maintenance inspections. Furthermore, the task allocation framework can be used to test or analyze different maintenance concepts or policies, as demonstrated in Subsection 3.5.2.

Future research on this work may consider the stochasticity associated with the TAP problem, or explore the uncertainty related to, e.g., the emerge of "non-routine tasks" or the aircraft utilization over the planning horizon. And this could enhance the robustness of the outcoming task execution plan. Furthermore, a stochastic approach could extend the current work to consider health prognostics and diagnostics, investigating the possibility of incorporating condition-based maintenance in the proposed framework. An alternative interesting future research direction is to integrate the maintenance check schedule optimizer with the task allocation framework proposed. And this could improve the overall quality of the maintenance plan, including checks schedule and task allocation per check.

REFERENCES

[1] M. M. D. Witteman, *A practical maintenance task packaging model applicable to aircraft maintenance*, Master's thesis (2019).

[2] S. P. Ackert, *Basics of Aircraft Maintenance Programs for Financiers,* (2010), (Accessed on September 28, 2017).

[3] A. Steiner, *A Heuristic Method for Aircraft Maintenance Scheduling under Various Constraints,* in *6th Swiss Transport Research Conference* (Monte Verità, Ascona, 2006).

[4] D. K. Friesen and M. A. Langston, *Variable sized bin packing,* SIAM Journal on Computing **15**, 222 (1986).

[5] A. Lodi, S. Martello, and M. Monaci, *Two-dimensional packing problems: A survey,* European Journal of Operational Research **141**, 241.

[6] C. Van Buskirk, B. Dawant, G. Karsai, J. Sprinkle, G. Szokoli, and R. Currer, *Computer-aided aircraft maintenance scheduling*, Tech. Rep. (Institute for Software-Integrated Systems, 2002).

[7] A. K. Muchiri and K. Smit, *Application of Maintenance Interval De-Escalation in Base Maintenance Planning Optimization,* Enterprise Risk Management **1** (2009), https://doi.org/10.5296/erm.v1i2.179.

[8] N. Hölzel, C. Schröder, T. Schilling, and V. Gollnick, *A Maintenance Packaging and Scheduling Optimization Method for Future Aircraft,* in *Air Transport and Operations Symposium* (2012).

3

[9] H. Li, H. Zuo, D. Lei, K. Liang, and T. Lu, *Optimal Combination of Aircraft Maintenance Tasks by a Novel Simplex Optimization Method,* Mathematical Problems in Engineering **2015** (2015), http://dx.doi.org/10.1155/2015/169310.

[10] D. S. Johnson, *Fast Algorithms for Bin Packing,* Journal of Computer and System Science **8**, 272 (1974).

[11] D. S. Johnson, *Fast Allocation Algorithms,* in *13th Annual Symposium on Switching and Automata Theory* (1972).

[12] I. Correia, L. Gouveia, and F. S. da Gama, *Solving the variable size bin packing problem with discretized formulations,* Computers & Operations Research **35**, 2103 (2008).

[13] J. Csirik, *An on-line algorithm for variable-sized bin packing,* Acta Informatica **26**, 697 (1989).

[14] J. Kang and S. Park, *Algorithms for the variable sized bin packing problem,* European Journal of Operational Research **147**, 365 (2003).

[15] M. Haouari and M. Serairi, *Heuristics for the variable sized bin-packing problem,* Computers & Operations Research **36**, 2877 (2009).

[16] S. Fazi, T. van Woensel, and J. C. Fransoo, *A Stochastic Variable Size Bin Packing Problem with Time Constraints,* `https://pure.tue.nl/ws/files/3678899/565767590859814.pdf` (2012), (Accessed on June 6, 2020).

[17] S. P. Ackert, *Aircraft Maintenance Handbook for Financiers,* `http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/aircraft_mx_handbook_for_financiers_v1.pdf` (2018).

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., USA, 1990).

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

4

Stochastic Aircraft Maintenance Check Scheduling Optimization

This chapter presents a stochastic model for aircraft maintenance check scheduling. Instead of knowing all aircraft daily utilization and maintenance check elapsed time during the entire planning horizon in the deterministic model, this information is only revealed the day after a current stage or after a maintenance check starts. To address the stochastic aircraft maintenance check scheduling, this chapter presents a lookahead approximate dynamic programming methodology, which infers the impact of a maintenance check decision using both future deterministic and stochastic demands. A real-life case study of a European airline shows that the proposed methodology can reduce the frequency of creating extra aircraft maintenance capacity while improving aircraft utilization compared with the estimation from the airline's planning approach.

The content of this chapter is based on the following research article:

Deng, Q. and Santos, B. F. (2021). Lookahead Approximate Dynamic Programming for Stochastic Aircraft Maintenance Check Scheduling Optimization. submitted to *European Journal of Operational Research*.

4.1. Introduction

Stochastic aircraft maintenance check scheduling (AMCS) is one of the next steps of the deterministic AMCS. Different from maintenance task allocation presented in Chapter 3, the stochastic AMCS still focuses on optimizing the aircraft maintenance check schedule. In addition to the detailed operation constraints in deterministic AMCS, the stochastic AMCS also considers the uncertainties. The uncertainties in AMCS mainly come from two sources: aircraft utilization and maintenance check elapsed time.

- Aircraft utilization is affected by weather conditions or flight disruption. In practice, bad weather conditions can shorten the life of some aircraft components or systems, and this further limits the time for commercial operations and force an aircraft to undergo maintenance earlier than planned. Flight disruptions, such as flight delays or flight cancellations, can also impact daily aircraft utilization, causing the deviation from the original aircraft maintenance plan.

- Despite flight disruptions, the maintenance schedule is affected by the elapsed time of maintenance checks. The maintenance planners allocate aircraft to maintenance slots on specific days for letter checks. One maintenance slot is one day of availability of a hangar for performing aircraft maintenance. The maintenance slots needed for a letter check are estimated based on the mean maintenance check elapsed time. For a specific maintenance check, the elapsed is usually uncertain in practice. It can be a few days longer than the estimation because of the non-routine maintenance tasks. These non-routine tasks include, e.g., replacement of major components (aircraft engines or landing gears), airworthiness directives [1], engineering orders [2], deferred tasks, non-scheduled maintenance tasks that result from faults, and additional maintenance need found when executing the routine task. The non-routines can be up to 50% of the workload performed during a maintenance check [3, 4]. 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.

Although Chapter 2 proposed a solution to the long-term deterministic AMCS, however, one of the limitations is that the optimization model described in Chapter 2 assumes complete information and does not include future uncertainty. The stochastic AMCS has not been tackled so far, not even adequately studied. Since it is impossible for airlines to follow a long-term aircraft maintenance schedule without adjustment in general, maintenance planners have to update the maintenance schedules from time to time due to flight disruptions or changes in maintenance tasks execution.

This chapter proposes a fast, short-term decision-making solution to cope with uncertainties and respond to changes in aircraft maintenance activities promptly, without comprising the long-term benefit. The research work described in this chapter is the continuation of our previous MPO solution extending the AMCS to a stochastic framework that considers uncertainty associated with aircraft daily utilization and maintenance check elapsed time. This chapter presents a lookahead approximate dynamic programming (ADP) methodology and uses it, for the first time, to address the stochastic AMCS. The contributions include:

- *Methodology*: The 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 additional maintenance slots for frequent light maintenance.

- *Application*: The proposed methodology is more robust than the previous deterministic approach present in the literature, in terms of fewer 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 the short-term aircraft letter check schedules within seconds without compromising the long-term efficiency of the solution.

4.2. STATE OF THE ART

Several publications address the aircraft maintenance related problems considering the stochastic elements. The earliest one can be traced back to 1966, Ref. [5] 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 [6], and the maintenance programs were process-orientated [7].

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. Ref. [8] 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 maintenance schedule and shorter computation time.

Ref. [9] was aware that airline planning models did not explicitly consider stochastic elements in operations, which often led to discrepancies between 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 determines 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, Ref. [9] still provides some insights on how random disruptions affect the daily operation of airlines and how airlines recover from disruptions. And this also prompts us to develop a dynamic optimal decision-making model for AMCS.

As mentioned in Ref. [9], stochastic simulation is a way of capturing uncertainty, particularly 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. Ref. [10] 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. And 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. Aircraft 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, Ref. [11] proposed a model enhancement (ME) algorithm for planning robust aircraft maintenance personnel rosters 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) and shown to provide robust and promising solutions. Following the path of Ref. [11], this chapter uses simulation to simulate aircraft utilization and maintenance elapsed time, which gives 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 Refs. [12–15], yet none of them deal with 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 Ref. [16]. 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.

4.3. NOMENCLATURE

AMCS Model Parameters:

d_k	Minimum interval between the start dates of two type k checks.
$\mathrm{fc}_{i,t}$	Average daily FC usage for aircraft i at day t
$\mathrm{fh}_{i,t}$	Average daily FH usage for aircraft i at day t
$I^i_{k\text{-DY}}$	Interval of type k check of aircraft i in terms of DY
$I^i_{k\text{FH}}$	Interval of type k check of aircraft i in terms of FH
$I^i_{k\text{-FC}}$	Interval of type k check of aircraft i in terms of FC
K	Collection of letter check type, $K = \{\text{A-check, B-check, C-check, D-check}\}$
N	Total number of aircraft
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
W	The set of all sample paths
ω_t	New information that arrives on day t
λ	Daily penalty for having an additional slot for type k check
π	Scheduling policy
ξ	A large number to prevent the waste of an available maintenance slot
γ	Discount factor

Other Parameters:

h	Hangar indicator
i	Aircraft indicator
k	Indicator for maintenance check type
t	Indicator of calendar day

Main Decision Variables:

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

State Related Decision Variables:

$a_{i,t}^k$	The attributes of aircraft i in the beginning of day t for type k check
$a_{i,t}$	The attributes of aircraft i in the beginning of day t
A_t	$A_t = \{a_{i,t} \mid i = 1, 2, \ldots, N\}$
$C_t(s_t, x_t)$	Contribution of choosing action x_t on s_t
$C_t^k(s_t, x_t)$	Contribution of choosing action x_t on s_t with respect to type k check
$\text{DY}_{i,t}^k$	Total DY of aircraft i in the beginning of day t for type k check
$\text{FC}_{i,t}^k$	Cumulative FC of aircraft i at t since last type k check
$\text{FH}_{i,t}^k$	Cumulative FH of aircraft i at t for type k check
$L_i\left(y_{i,t}^k\right)$	Mean estimated elapsed time of next check with label $y_{i,t}^k$ of aircraft i
$M_{h,t}^k$	Binary variable to indicate if type k check can be performed in hangar h on day t
M_t^k	Hangar capacity of type k check, $M_t^k = \sum_h M_{h,t}^k$
s_t^k	State variable with respect to type k check
s_t	Pre-decision state variable, $s_t = \{s_t^k\}_{k\in K}$
$\hat{s}_t$	Post-decision state variable before new information arrives
$R_{i,t}^k$	Remaining utilization of aircraft i before the next type k check
$y_{i,t}^k$	Next maintenance label for of type k check of aircraft i on day t
$z_{i,t}^k$	The end date of type k check of aircraft i
$\delta_{i,t}^k$	Binary variable to indicate if aircraft i is undergoing type k check on day t
$\eta_{i,t}^k$	Binary variable to indicate if aircraft i needs an extra slot of type k check on day t
Ψ	$\Psi \in \{\text{FH}, \text{FC}\}$
ψ	$\Psi \in \{\text{fh}, \text{fc}\}$
$\mathcal{S}^X(s_t, x_t)$	Transition function from s_t to $\hat{s}_{t+1}$, $\hat{s}_{t+1} = \mathcal{S}^X(s_t, x_t)$ before arrival of new information
$\mathcal{S}^W(s_t, \omega_t)$	Transition function from $\hat{s}_t$ to s_t, $s_t = \mathcal{S}^W(\hat{s}_t, \omega_t)$ when the new information is known
$V_t(s_t)$	The value of being in a state s_t

4.4. PROBLEM FORMULATION

The AMCS problem has already been presented in Chapter 2. This chapter adopts the same definition of maintenance interval and assumptions. The nomenclature and corresponding description can be found in 4.3.

4.4.1. STATE TRANSITION IN STOCHASTIC AMCS

The state vector s_t is a set of attributes that influence our decisions, and this set also includes available maintenance slots of each check type:

$$s_t = \left\{\left\{s_t^k\right\} \middle| k \in K\right\}, \quad K \in \{\text{A-check, B-check, C-check, D-check}\}, \quad s_t^k = \left\{\left\{a_{i,t}^k\right\}_{i=1}^N\right\} \tag{4.1}$$

where, each $a_{i,t}^k$ contains the information of aircraft i on day t of check type k:

$$a_{i,t}^k = \Big\{\underbrace{M_t^k, z_{i,t}^k(\omega_t), \delta_{i,t}^k, \eta_{i,t}^k}_{\text{Type 1}}, \underbrace{\mathrm{DY}_{i,t}^k, \mathrm{FH}_{i,t}^k, \mathrm{FC}_{i,t}^k, y_{i,t}^k}_{\text{Type 2}}, \tag{4.2}$$

$$\underbrace{L_i\left(y_{i,t}^k\right), \mathrm{fh}_{i,t}, \mathrm{fc}_{i,t}, \Delta L_i^\omega\left(y_{i,t}^k\right), \Delta\mathrm{fh}_{i,t+1}^\omega, \Delta\mathrm{fc}_{i,t+1}^\omega}_{\text{Type 3}}\Big\} \tag{4.3}$$

Table 4.1: Different types of attribute within a state s_t.

Type 1 $a_{i,t}^{(1),k}$	Attributes at time t that impact the action x_t and are modified when there is new information or after a maintenance check starts
Type 2 $a_{i,t}^{(2),k}$	Attributes at time t that are updated every time based on their value at time $t-1$
Type 3 $a_{i,t}^{(3),k}$	Attributes at time t that depend on exogenous information and can be estimated according to historical aircraft utilization and maintenance data

These attributes can be divided into three types, as showed in Table 4.1, and the uncertainties come from the attributes of Type 3, the aircraft utilization, and maintenance check elapsed time. For aircraft utilization, maintenance planners of airlines 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. Even so, in the model formulation, we use the average value based on the historical daily utilization of fleet and elapsed time of letter checks. To serve our purpose for the stochastic AMCS, we adapt the post-decision state vector $\hat{s}_{t+1}$ before the arrival of new information:

$$\hat{s}_t = \mathcal{S}^X\left(s_t, x_t\right) \tag{4.4}$$

where $\mathcal{S}^X$ denote the state transition function without knowing any new information. In stochastic AMCS, this chapter assumes that the new information $\{\omega_t\}_{t=t_0+1}^{T+1}$ is revealed when a letter check starts or an aircraft ends its daily operation, then it updates $\hat{s}_t$:

$$s_{t+1} = \mathcal{S}^W\left(\hat{s}_t, \omega_{t+1}\right) \tag{4.5}$$

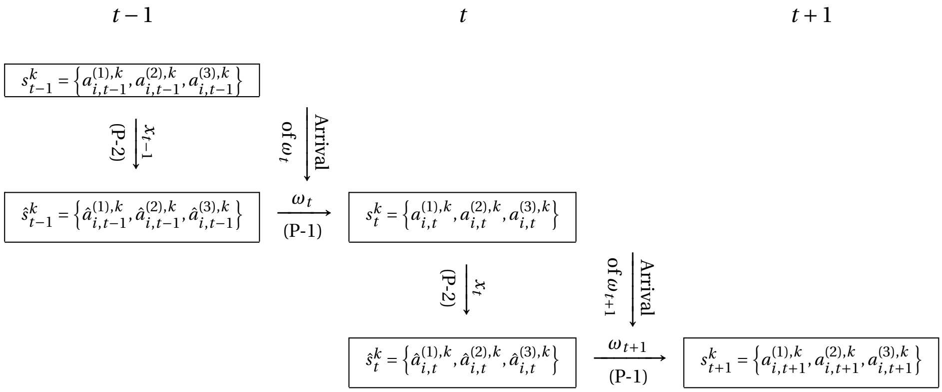


Figure 4.1: A two-phase attribute update mechanism: Phase 1 (P-1) updates the set of pre-decision attributes $\hat{s}_t^k$ to s_t^k before defining any action; after performing an action x_t, Phase 2 (P-2) updates s_t^k to $\hat{s}_{t+1}^k$.

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

$$\left(\hat{s}_{t_0-1}, \omega_{t_0}, s_{t_0}, x_{t_0}, \hat{s}_{t_0}, \omega_{t_0+1}, \ldots, s_{t-1}, x_{t-1}, \hat{s}_{t-1}, \omega_t, s_t, \ldots, s_T, x_T, \hat{s}_T, \omega_{T+1}, s_{T+1}, \ldots\right) \tag{4.6}$$

The reason of including the post-decision state $\hat{s}_{t_0-1}$ as the initial state and initial information ω_{t_0} in (4.6) 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:

$$\begin{cases} s_t = \mathcal{S}^W\left(\hat{s}_{t-1}, \omega_t\right) \\ x_t = \mathcal{X}^\pi(s_t) \\ \hat{s}_t = \mathcal{S}^X\left(s_t, x_t\right) \end{cases} \quad \text{for} \quad t = t_0,\ t_0+1,\ \ldots,\ T \tag{4.7}$$

As shown in Figure 4.1, the state transition updates the attributes over the time horizon in two phases: pre-decision (Phase 1) and post-decision (Phase 2). The new information, ω_t, arrives at the beginning of day t. The pre-decision phase (before making any new decision) renews the hangar capacity, aircraft availability, and utilization for time t. It updates either the elapsed time or updates the aircraft utilization based on actual FH and FC (according to the new information ω_t). 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 be finished. 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.

UPDATE OF TYPE 1 ATTRIBUTES

In Phase 1 (pre-decision phase in Figure 4.1), we first check if t is the end day for an ongoing aircraft check before any action, or give the actual end date of a type k check if it starts at $t-1$ (we assume the actual elapsed time is only known when the check starts, namely, the new information arrives at t), for all aircraft:

$$z_{i,t}^{k}(\omega_t) = \begin{cases} 0 & \text{if } \hat{z}_{i,t-1}^{k} = t-1 \\ \hat{z}_{i,t-1}^{k} + \Delta L_i^{\omega}(y_{i,t-1}^{k}) & \text{if } \chi_{i,t-1}^{k} = 1 \\ \hat{z}_{i,t-1}^{k} & \text{otherwise} \end{cases} \tag{4.8}$$

where $\Delta L_i^{\omega}\left(y_{i,t-1}^{k}\right)$ follows a certain distribution and its value depends on the realization ω_t. If the end date of a type k check for an aircraft i is larger than the current calendar day t, it means the check is still ongoing. Therefore we update $\hat{\delta}_{i,t-1}^{k}$ to $\delta_{i,t}^{k}$:

$$\delta_{i,t}^{k} = \begin{cases} 0 & \text{if } z_{i,t}^{k} = 0 \\ \hat{\delta}_{i,t-1}^{k} & \text{otherwise} \end{cases} \tag{4.9}$$

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

$$M_t^k = \sum_h M_{h,t}^k - \sum_{i=1}^{N} \delta_{i,t}^{k} \tag{4.10}$$

where $M_{h,t}^k$ is the maintenance capacity per hangar h at time t. The the number of additional slots of type k check, $\eta_{i,t}^k$, is updated according to the current capacity $M_{h,t}^k$:

$$\eta_{i,t}^{k} = -\min\left\{0, M_t^k\right\} \tag{4.11}$$

In Phase 2 (post-decision phase in Figure 4.1), the action x_t is taken into account to update Type 1 attributes. For all aircraft that start type k check on day t ($\chi_{i,t}^k = 1$), the values of $z_{i,t}^k$ and $\delta_{i,t}^k$ need to be updated. The $z_{i,t}^k$ is updated according to:

$$\hat{z}_{i,t}^{k} = \begin{cases} t + L_i\left(y_{i,t}^{k}\right) & \text{if } \chi_{i,t}^{k} = 1 \\ z_{i,t} & \text{otherwise} \end{cases} \tag{4.12}$$

Note that $L_i(y_{i,t}^k)$ is the mean elapsed time according to historical maintenance check data. Following this update, the values of $\delta_{i,t}^k$ can also be renewed:

$$\hat{\delta}_{i,t}^{k} = \begin{cases} 1 & \text{if } \chi_{i,t}^{k} = 1 \\ \delta_{i,t}^{k} & \text{otherwise} \end{cases} \tag{4.13}$$

UPDATE OF TYPE 2 ATTRIBUTES

Once the action 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:

$$\mathrm{DY}^k_{i,t+1} = \left(1-\delta^k_{i,t}\right)\left(\mathrm{DY}^k_{i,t}+1\right) \tag{4.14}$$

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

$$\Psi^k_{i,t+1} = \left(1-\delta^k_{i,t}\right)\left(\Psi^k_{i,t}+\left(1-\delta^{k'}_{i,t}\right)\left[\psi_{i,t}+\Delta\psi_{i,t+1}(\omega_{t+1})\right]\right),\ \Psi\in\{\mathrm{FH},\mathrm{FC}\},\ \psi\in\{\mathrm{fh},\mathrm{fc}\} \tag{4.15}$$

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 was scheduled in the previous time step (i.e., $\delta^k_{i,t}=1$). 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., $\delta^{k'}_{i,t}=1$). $\Delta\psi_{i,t}(\omega_t)$ follows a certain distribution and $\psi_{i,t}$ is the mean daily utilization of aircraft i according to airline estimation.

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:

$$y^k_{i,t+1} = \begin{cases} y^k_{i,t}+1 & \text{if } \chi^k_{i,t}=1 \\ y^k_{i,t} & \text{otherwise} \end{cases} \tag{4.16}$$

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:

- $L_i\left(y^k_{i,t}\right)$ is the mean elapsed time from historical maintenance data.
- $\mathrm{fh}_{i,t}$ and $\mathrm{fc}_{i,t}$ are estimated according to historical aircraft FH and FC.
- $\Delta L^\omega_i\left(y^k_{i,t}\right)$, $\Delta\mathrm{fh}^\omega_{i,t+1}$ and $\Delta\mathrm{fc}^\omega_{i,t+1}$ follow certain distributions respectively, and their values all depend on the realization of ω_{t+1}. We assume that the new information ω_{t+1} arrives on day $t+1$.

4.4.2. 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:

$$\mathrm{DY}^k_{i,t}+1 \le I^i_{k\text{-DY}} \tag{4.17}$$

$$\Psi^k_{i,t}+\psi_{i,t} \le I^i_{k\text{-}\Psi} \tag{4.18}$$

where $\Psi \in \{\text{FH}, \text{FC}\}$ and $\psi \in \{\text{fh}, \text{fc}\}$. 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.

Before instigating an action, we need to verify whether or not there are sufficient maintenance slots for a type k check in one of the hangars during the entire mean maintenance elapse time $L_i\left(y_{i,t}^k\right)$:

$$\chi_{i,t}^k \le \frac{\sum_{\tau=t}^{t+L_i\left(y_{i,t}^k\right)} M_{h,\tau}^k}{L_i\left(y_{i,t}^k\right)}, \; k \in K, \; t \in [t_0, T], \; h \in \{\text{the set of hangars for type } k \text{ check}\} \tag{4.19}$$

$L_i\left(y_{i,t}^k\right)$ is estimated according to historical data. Note that the actual maintenance elapsed time of a type k check can be larger than $L_i\left(y_{i,t}^k\right)$, if additional slots are needed for an ongoing check, they will be created and updated according to (4.11).

Some airlines require a minimum number of days (d_k) between the start dates of two type k checks 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 $[t, t+d_k)$

The requirement for the start date can be translated in the following equations:

$$\sum_{i=1}^{N} \chi_{i,t}^k \le \begin{cases} 1 & \text{if } d_k > 0 \text{ and } \sum_{i=1}^{N} \chi_{i,\tau}^k = 0, \; \forall \tau \in [t-d_k, t) \\ M_t^k & \text{otherwise} \end{cases} \qquad t \in [t_0, T] \tag{4.20}$$

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

4.4.3. Objective Function

We use the same objective function as described in [16], minimizing the unused FH [17, 18], instead of the total maintenance cost, due to the following reasons:

- The available maintenance cost data is unreliable and hard to associate to 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.

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

$$C_t^k\left(s_t^k, x_t^k\right) = \chi_{i,t}^k\left(I_{k\text{-FH}}^i - \mathrm{FH}_{i,t}^k\right), \quad k \in K \tag{4.21}$$

The *contribution function* of FH loss on day t is calculated by:

$$C_t\left(s_t, x_t\right) = \sum_{k \in K}\sum_{i=1}^{N}\left[C_t^k\left(s_t^k, x_t^k\right) + \lambda\eta_{i,t}^k\right] \tag{4.22}$$

where the first term on the right-hand side 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 very costly, and it should only be an option if it avoids an aircraft on the ground, waiting for a maintenance slot. For that reason, the value of λ_t should be much larger than $C_t^k\left(s_t^k, x_t^k\right)$.

Our objective is then 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 $T - t_0$:

$$\min_{\pi} \mathbb{E}\left\{\sum_{t=t_0}^{T}\gamma^{t-t_0}C_t\left(s_t, \mathscr{X}^{\pi}\left(s_t\right)\right)\Big| s_{t_0}\right\} \tag{4.23}$$

where π is the scheduling policy that generates actions based on s_t, s_{t_0} denotes the initial state. $\mathscr{X}^{\pi}\left(s_t\right)$ is the set of actions associated with s_t if the policy π is adopted.

4.4.4. OPTIMIZATION MODEL

After the introduction of state transition, constraints, and objective function, the optimization problem is to minimize (4.23), subject to constraints (4.8)–(4.20).

The optimal maintenance check scheduling policy over the time horizon $[t_0, T]$ can be found by recursively computing the Bellman's equation:

$$V_t(s_t) = \min_{x_t}\left\{C_t\left(s_t, x_t\right) + \gamma\sum_{s_{t+1}} p\left(s_{t+1}\middle| s_t, x_t\right)V_{t+1}\left(s_{t+1}\right)\right\} \tag{4.24}$$

where $s_{t+1} = \mathcal{S}^W\left(\hat{s}_t, \omega_{t+1}\right) = \mathcal{S}^W\left(\mathcal{S}^X\left(s_t, \mathscr{X}^{\pi}\left(s_t\right)\right), \omega_{t+1}\right)$, and $p\left(s_{t+1}\middle| s_t\right)$ is the probability of transitioning from state s_t to state s_{t+1}. The Bellman's equation expresses the value of being at each state s_t, by considering the immediate contribution of an action x_t and the future value.

4.5. METHODOLOGY

The stochastic AMCS is a typical Markov Decision Process (MDP) consisting of four elements: a set of states $\{s_t\}$, a set of associated actions $\left\{x_t | x_t = \mathscr{X}^{\pi}\left(s_t\right)\right\}$, the immediate reward of doing an action $C_t\left(s_t, x_t\right)$ and the probability $p\left(s_{t+1}|s_t\right)$ of transition from a state

s_t to another new state s_{t+1}. Here we use Figure 4.2 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 corresponding actions s_{t_0}. 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 at this moment. The new information ω_{t_0+1} is a stochastic variable, each realization $\omega^l_{t_0+1}$ 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 $\omega^l_{t_0+1}$. However, $\omega^l_{t_0+1}$ is only revealed after an action is made. In order to know the final optimal state we have to keep making decisions until we reach the end of planning horizon.

There are three main hindrances that prevent us from planning the optimal schedule:

H.1 Multi-dimensional state vector s_t (each aircraft has many attributes)

H.2 Multi-dimensional action vector x_t (selecting different combinations of aircraft for maintenance check)

H.3 Very large outcome space (the optimal final state is unknown)

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

$$\prod_{k\in K}\sum_{m_k=0}^{M^k_t}\frac{N!}{(N-m_k)!\,m_k!} \tag{4.25}$$

where $\frac{N!}{(N-m_k)!\,m_k!}$ represents the possible selections of aircraft for type k check. The number of outcome states for type k check is the same as (4.25). As a result, the number of possible actions on the day T is:

$$\prod_{t=t_0}^{T}\prod_{k\in K}\sum_{m_k=0}^{M^k_t}\frac{N!}{(N-m_k)!\,m_k!} \tag{4.26}$$

Even if 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 problem formulated as MDP is dynamic programming (DP). However, a classic implementation DP requires solving (4.24) recursively, from T to t_0. This process is not possible for AMCS since the final state is unknown. Thus, forward induction becomes the only option, meaning that we have to determine optimal actions from an initial state until the end of the planning horizon while estimating the impacts of current actions on future stages.

Chapter 2 addresses the deterministic AMCS optimization using a DP-based methodology by defining maintenance check priority, applying a thrifty algorithm to estimate if the remaining slots will be sufficient, discretization, and state aggregation under DP framework [16]. However, the DP-based methodology is not suitable for stochastic AMCS since it relies on having deterministic information on aircraft daily utilization and maintenance elapsed time. The DP-based methodology keeps a set of *workable* states for

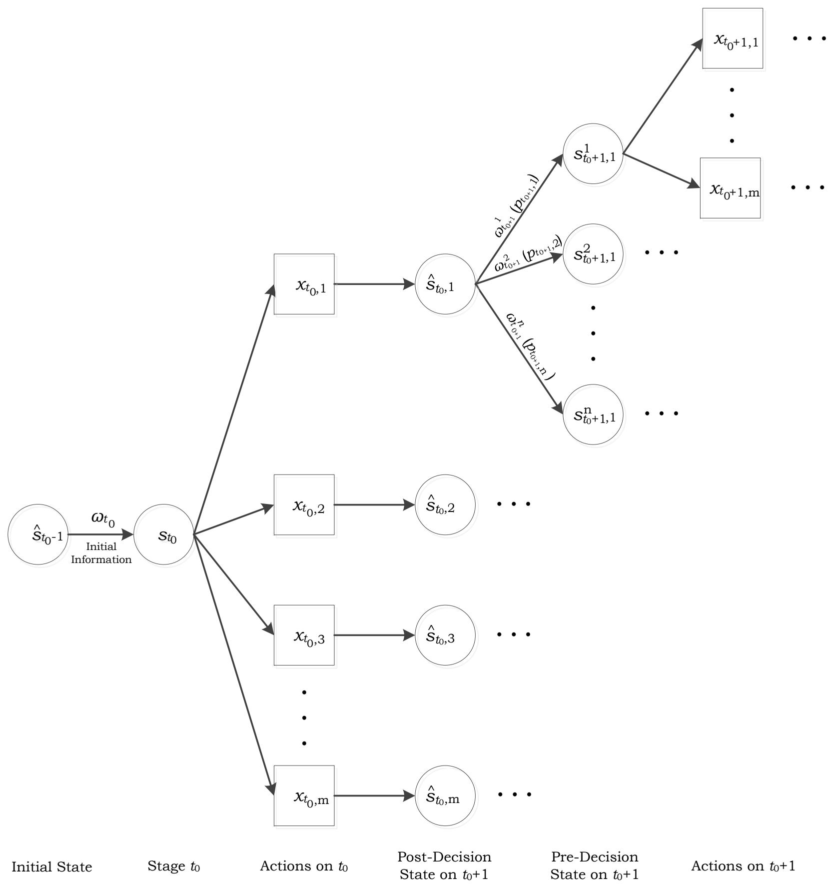


Figure 4.2: An example of state transition from stage t_0 to stage t_0+1 in stochastic AMCS. $\left\{x_{t_0,j}\right\}$ is the set of possible actions associated with s_{t_0} and $\left\{\hat{s}_{t_0,j}\right\}$ is the set of resulting post-decision states. The pre-decision state $s^j_{t_0+1,1}$ is only 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 $s^j_{t_0+1,1}$.

each day t using discretization and aggregation, from which it computes the *workable* states for $t+1$. But in stochastic AMCS, once we make a maintenance decision on t, there is only one state on $t+1$ after the new information is revealed. Working with a set of *workable* states and exploring the optimal sequence of actions is no longer possible. Therefore, we propose an approximate dynamic programming (ADP) methodology in which we approximate the future costs of performing a maintenance check action under the DP framework, using Monte Carlo simulation to capture the uncertainties in our decision framework.

This section presents the detail of the lookahead ADP methodology for stochastic AMCS. We begin with a brief introduction to the ADP concept in Subsection 4.5.1. Subsection 4.5.2 presents how we use Monte Carlo sampling to simulate uncertainty. Subsection 4.5.3 defines maintenance check priority for each aircraft and Subsection 4.5.4 defines basic rules for AMCS. After that, we describe two reference AMCS policies in Subsection 4.5.5 as benchmarks. In Subsection 4.5.6, we present the detail of the lookahead ADP methodology. The last subsection (Subsection 4.5.7) shows an analysis of algorithm complexity.

4.5.1. 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 [19]. 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 main idea is to make decisions by optimizing instant reward/cost (*myopic* policy); or look ahead to future reward/cost (*lookahead* policy) to make decisions; or use approximation techniques, such as computer simulation and machine learning to approximate either the optimal policy (policy iteration) or the value function (value iteration), instead of solving (4.24). Policy iteration [20, 21] or value iteration [22–25] usually requires a model, either parametric or non-parametric, to capture the features of a state. One common approach is to formulate the policy/value function as a linear combination of the values from each feature. However, neither policy iteration nor value iteration is deemed suitable for stochastic AMCS due to the following limitations:

1) *Objective Setting*: Since our objective is to minimize the total unused FH between successive maintenance checks, there is no direct link between policy/objective function and the features of a state that can impact the maintenance check decision. Moreover, the total unused FH is the summation of unused FH of each letter check for the entire planning horizon; however, the number of letter checks for each aircraft is a part of the solution.

2) *Changing of Fleet Size*: The fleet size varies during the planning horizon, e.g., new aircraft may phase-in, and old aircraft can phase-out/retire, which changes the environment conditions and compromises the training of policy function or value function.

On the other hand, using the *lookahead* approach for ADP instead of policy iteration or value iteration can skip the steps of feature selection and training of policy/value function. The key to *lookahead* approach is a *lookahead* policy that estimates the cost of performing a maintenance check action from a future period. Besides, a *lookahead* policy can easily adapt itself to the new fleet status and environment, e.g., the change of fleet size or maintenance slots. Therefore, this chapter resorts to implementing a *lookahead* policy under the ADP, which combines Monte Carlo sampling and simulation for the stochastic AMCS. It uses the estimation of the cost of performing a maintenance check action to approximate the value $V_{t+1}(s_{t+1})$ in (4.24). Based on the estimation of future cost and immediate contribution, it makes the best maintenance check decision.

4.5.2. MODELING OF UNCERTAINTY

4

Inspired by [9] and [10], we use stochastic simulation to capture uncertainty (generate information). Monte Carlo sampling is a computational technique based on constructing a random process and carrying out a numerical experiment by N-fold sampling from a random sequence of numbers with a prescribed probability distribution [26]. The sets of sample paths $\{w^n\}$, or so-called new information, are generated by Monte Carlo sampling. Each sample path is a sequence of information $w^n = \left\{\omega^n_{t_0+1}, \omega^n_{t_0+2}, \dots, \omega^n_{T+1}\right\}$. 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 $\Delta\text{fh}^{\omega}_{i,t}$ from normal distribution $N\left(\mu_i, \sigma_i^2\right)$, and $\Delta\text{fc}^{\omega}_{i,t}$ also follows the same process.

On the other hand, for the maintenance elapsed time, we opt to generate integers to be consistent with the daily maintenance slots and flight schedule. We have adapted the Monte Carlo sampling to serve our purpose. Given a set of historical maintenance check elapsed time from an airline:

Step 1: Count the number of data points for a specific letter check (n_{count}).

Step 2: Generate a uniformly distributed integer q, that is, $q \sim U(1, n_{\text{count}})$.

Step 3: Pick the q^{th} data point as the additional maintenance check elapsed time (new information).

We use an example to elaborate Monte Carlo sampling for maintenance check elapsed time. Given a set of C-check label and extra elapsed time (in working days) of aircraft i:

$$\textbf{C1.1:} \quad -1,\ 0,\ 1,\ -2,\ 2,\ 0,\ 0,\ 0,\ 0,\ 2,\ -1,\ 0,\ 0,\ 0,\ 1,\ -1 \tag{4.27}$$

where "-1" means C1.1 finishes one day earlier, "-2" means C1.1 ends two day earlier, "1" indicates that it takes one more day than expected, and "2" indicates that C1.1 lasts two days longer than average. We observe from (4.27) that there are 16 data points in total $(n_{\text{count}} = 16)$, these 16 points represents the historical duration of C1.1. Next, we generate a uniformly distributed integer $q \in [1, 16]$, then pick the q^{th} data point as the actual elapsed time of **C1.1**, and that completes the sampling of one specific check. For example, if $q = 5$, we pick the 5^{th} data point from (4.27) and set $\Delta L^{\omega}_i\left(y^k_{i.t}\right) = 2$, where $y^k_{i.t} = $ C1.1. It repeats the same process for all letter check labels for the entire fleet.

After Monte Carlo sampling, the new information ω_{t+1} has the form of:

$$\omega_{t+1} = \{\omega^{\mathrm{A}}_{t+1}, \omega^{\mathrm{B}}_{t+1}, \omega^{\mathrm{C}}_{t+1}, \omega^{\mathrm{D}}_{t+1}\} \tag{4.28}$$

$$\omega^k_{t+1} = \left\{\Delta L^{\omega}_i\left(y^k_{i,t}\right),\ \Delta\mathrm{fh}^{\omega}_{i,t+1},\ \Delta\mathrm{fc}^{\omega}_{i,t+1}\right\} \qquad t \in [t_0+1, T],\ k \in \{\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D}\} \tag{4.29}$$

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 (policies), and we call this process *one simulation*.

4.5.3. DEFINING MAINTENANCE CHECK PRIORITY

Another major challenge in stochastic AMCS is the multi-dimensional action vector. According to (4.25), there are $\sum_k \sum_{m_k=0}^{M^k_t} \prod_k \frac{N!}{(N-m_k)!\ m_k!}$ actions on day t. To reduce the number of maintenance check actions, we propose a prioritization solution in the previous work [16], 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 [16], we use the term *expected remaining utilization* in stochastic 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:

$$R^k_{i,t} = \min\left\{R^{k\text{-DY}}_{i,t},\quad R^{k\text{-FH}}_{i,t},\quad R^{k\text{-FC}}_{i,t}\right\} \tag{4.30}$$

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

$$R^{k\text{-DY}}_{i,t} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{r \le I^i_{k\text{-DY}} - \mathrm{DY}^k_{i,t}\right\} \tag{4.31}$$

$$R^{k\text{-}\Psi}_{i,t} = \underset{r\in\mathbb{N}}{\operatorname{argmax}}\left\{\sum_{\tau=t}^{t+r} \mathrm{fh}_{i,\tau} \le I^i_{k\text{-}\Psi} - \Psi^k_{i,t}\right\} \tag{4.32}$$

where $\Psi \in \{\mathrm{FH},\ \mathrm{FC}\}$, $\psi \in \{\mathrm{fh},\ \mathrm{fc}\}$, $\psi_{i,\tau}$ and $\mathrm{fc}_{i,\tau}$ 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 $\left\{R^k_{i,t}\right\}_{i=1}^{N}$ in ascending order:

$$\tilde{R}^k_{1,t},\ \tilde{R}^k_{2,t},\ \tilde{R}^k_{3,t},\ \ldots,\ \tilde{R}^k_{N,t} \qquad \tilde{R}^k_{i,t} \le \tilde{R}^k_{i+1,t},\ \tilde{R}^k_{i,t} \in \left\{R^k_{i,t}\right\}_{i=1}^{N} \tag{4.33}$$

The fleet is scheduled maintenance check according to the sequence in (4.33): aircraft with lower *expected remaining utilization* is given a higher check priority. For each letter check type, after assigning priorities to the entire fleet, the combination of aircraft selection for maintenance and the number of outcome states of each action is reduced from (4.25) 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 other check types.

4.5.4. BASIC SCHEDULING RULES FOR STOCHASTIC AMCS

This chapter defines some basic rules for making AMCS decisions before presenting the scheduling policies. These basic rules are the prerequisites for the stochastic AMCS:

(i) An aircraft i is allocated a type k check if its *expected remaining utilization* is lower than a threshold (i.e., when $R^k_{i,t} \le R^k_{\text{lb}}$). 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 that has lowest *expected remaining utilization* $\tilde{R}^k_{1,t} = \min_i \left\{R^k_{i,t}\right\}$ has highest priority of type k check.

(iii) If aircraft i has a higher type k check priority than aircraft j $\left(R^k_{i,t} < R^k_{j,t}\right)$ 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 in the AMCS 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

4.5.5. REFERENCE SCHEDULING POLICIES

To address the stochastic 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 the most fundamental scheduling policies, although they are not common in the AMCS application. The results from the *myopic* policy indicate the extreme of being greedy. It also 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 results from the *thrifty* policy indicate the extreme of being conservative. It provides a lower bound for the average aircraft utilization and an upper bound for the total number of maintenance checks. In this study, we use the outcomes from *myopic* and *thrifty* policies to benchmark our hybrid lookahead policy so that we can have some insight into how far the KPIs are from the associated lower and upper bounds.

MYOPIC POLICY

Myopic policy is one of the most elementary policies. It requires only the state s_t and 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 decision 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 (4.24):

$$x^*_t = \underset{x_t \in X_t}{\operatorname{argmin}} \{C_t(s_t, x_t)\} \tag{4.34}$$

where X_t denotes the set of possible actions of day t, $X_t = \left\{\mathscr{X}^\pi(s_t)\right\}$. The *myopic* policy runs very fast and if it results in no additional slot in stochastic AMCS (e.g., there is sufficient aircraft maintenance capacity), then (4.34) 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.

THRIFTY POLICY

The *thrifty* policy is a conservative policy that schedules maintenance check whenever there is an available slot [16], and this can be interpreted as:

$$x_t^* = \underset{x_t \in X_t}{\operatorname{argmax}} \left\{ \sum_{k \in \{A,C\}} \sum_{i=1}^{N} \chi_{i,t}^k \middle| \sum_{i=1}^{N} \chi_{i,t}^k \leq \sum_h M_{h,t}^k,\ \chi_{i,t}^k \in x_t \right\} \tag{4.35}$$

where $X_t = \{\mathscr{X}^\pi(s_t)\}$. Similar to the *myopic* policy, the *thrifty* policy only requires the state s_t to make a maintenance check decision without looking into the future cost. In particular, it checks whether or not 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

4.5.6. 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 [16]. The hybrid lookahead policy combines deterministic and stochastic forecasts.

A HYBRID LOOKAHEAD POLICY

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

$$x_t^* = \underset{x_t \in X_t}{\operatorname{argmin}} \left\{ C_t(s_t, x_t) + \gamma \overline{V}_t(s_t) \right\} \tag{4.36}$$

where $\overline{V}_t(s_t)$ is an approximation of the value function $V_t(s_t)$ in (4.24) and also the key to solve (4.36). 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. In this way, we first use the *thrifty* policy discussed in [16] to explore the future and estimate the number of additional maintenance slots that would be needed if an action is taken:

$$g_k(\hat{s}_t, t+t_h) = \sum_{i=1}^{N} \sum_{\tau=t}^{t+t_h} \hat{\eta}_{i,\tau}^k, \quad k \in K \tag{4.37}$$

where $\hat{\eta}_{i,\tau}^k$ 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 (4.37) 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 (4.35) from $t+1$ to $t+t_h$.

As C-checks happen every 18–24 months, and D-checks occur every 5–6 years, the deviation of aircraft daily utilization from its mean value will cancel out during this long period. It means that we can use the average daily utilization of each aircraft to simulate when the coming C-/D-checks take place. The “self-cancellation” of uncertainty also applies to maintenance check elapsed time. For instance, one C-check may require several days more to complete, but the other needs fewer days to finish. If these two C-checks are executed one after another in the same hangar, it can mitigate the impact of uncertainty. However, (4.37) it 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 (4.37), 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 optimal C- & D-check actions using deterministic forecasts
- *Step 2*: Determine the optimal A- & B-check actions using stochastic forecasts

DETERMINE 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, if we skip a maintenance slot on the day t_1, it 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 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, $\exists i,\ R^k_{i,t} \le R^k_{\text{lb}}$ and constraint (4.19) 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., $\sum_{i=1}^{N} \chi^k_{i,t} = 0$

According to this logic, we use the following approximation for $V_t(s_t)$ in (4.24):

$$V_t(s_t) \approx \overline{V}^{(1)}_t(s_t) = \sum_k \Bigg(\lambda g_k(\hat{s}_t, t+t_h) + \underbrace{\max_{R^k_{i,t} \le R^k_{\text{lb}}} \left\{ \text{sgn}\left(\frac{\sum_{\tau=t}^{t+L_i\left(y^k_{i,t}\right)} M^k_{h,\tau}}{L_i\left(y^k_{i,t}\right)} - \chi^k_{i,t} \right) \right\}}_{\textbf{C.1}} \underbrace{\text{sgn}\left(g_k(\hat{s}_t, t+t_h)\right)}_{\textbf{C.2}} \underbrace{\left[1 - \text{sgn}\left(\sum_{i=1}^{N} \chi^k_{i,t}\right)\right]}_{\textbf{C.3}} \xi \Bigg) \tag{4.38}$$

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:

$$\text{sgn}(\alpha) = \begin{cases} -1 & \text{if } \alpha < 0 \\ 0 & \text{if } \alpha = 0 \\ 1 & \text{if } \alpha > 0 \end{cases} \tag{4.39}$$

We decide C- and D-check actions for the day t, i.e., only keep the optimal C- and D-check actions:

$$x^*_{t,\text{det}} = \left\{x^{A*}_{t,\text{det}}, x^{B*}_{t,\text{det}}, x^{C*}_{t,\text{det}}, x^{D*}_{t,\text{det}}\right\} = \underset{x_t \in X_t}{\text{argmin}} \left\{C_t(s_t, x_t) + \gamma \overline{V}^{(1)}_t(s_t)\right\} \tag{4.40}$$

$$x^{C*}_t = x^{C*}_{t,\text{det}}, \quad x^{D*}_t = x^{D*}_{t,\text{det}} \tag{4.41}$$

In (4.40), "det" is the abbreviation of "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 (4.40) and (4.41), we fix x^{C*}_t and x^{D*}_t.

4

DETERMINE 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 $[t+1, t+t_l]$ ($t_l \ll t_h$). We carry out Monte Carlo simulations:

$$w^n_{t+1} = \left\{\omega^n_{t+1}, \omega^n_{t+2}, \ldots, \omega^n_{t+t_l+1}\right\}, \quad n = 1, 2, \ldots, n_{\text{sample}}, \ t \in [t_0, T] \tag{4.42}$$

$$g^{\omega}_k\left(\hat{s}_t, t+t_l, w^n_{t+1}\right) = \sum_{i=1}^{N} \sum_{\tau=t}^{t+t_l} \eta^k_{i,\tau}\left(\omega^n_{\tau+1}\right), \quad k \in K \tag{4.43}$$

$$G_k(\hat{s}_t, t+t_l) = \frac{1}{n_{\text{sample}}} \sum_{n=1}^{n_{\text{sample}}} g^{\omega}_k\left(\hat{s}_t, t+t_l, w^n_{t+1}\right) \tag{4.44}$$

Similar to (4.38), we use the following approximation for $V_t(s_t)$ in (4.24):

$$V_t(s_t) \approx \overline{V}^{(2)}_t(s_t) = \sum_k \left(\lambda G_k(\hat{s}_t, t+t_l) + \max_{R^k_{i,t} \le R^k_{\text{lb}}} \left\{ \text{sgn}\left(\frac{\sum_{\tau=t}^{t+L_i\left(y^k_{i,t}\right)} M^k_{h,\tau}}{L_i\left(y^k_{i,t}\right)} - \chi^k_{i,t} \right) \right\} \text{sgn}\left(G_k(\hat{s}_t, t+t_l)\right) \left[1 - \text{sgn}\left(\sum_{i=1}^{N} \chi^k_{i,t}\right)\right] \xi \right) \tag{4.45}$$

After that, we determine the optimal A- and B-check actions:

$$x^*_t = \{x^{A*}_t, x^{B*}_t, x^{C*}_t, x^{D*}_t\} = \underset{\{x^A_t, x^B_t, x^{C*}_t, x^{D*}_{t,}\} \in X_t}{\text{argmin}} \left\{C_t\left(s_t, \{x^A_t, x^B_t, x^{C*}_t, x^{D*}_{t,}\}\right) + \gamma\lambda \overline{V}^{(2)}_t(s_t)\right\} \tag{4.46}$$

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 (4.40)–(4.46) a lookahead ADP methodology. The detail of lookahead ADP methodology is presented in Algorithm 3.

Algorithm 3 A Lookahead ADP Methodology for Stochastic AMCS Optimization

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

procedure APPROXIMATE DYNAMIC PROGRAMMING
  $t \leftarrow t_0$

  while $t_0 < T$ **do**
    $\omega_t \longleftarrow \left\{\Delta L_i^{\omega}\left(y_{i,t-1}^k\right), \Delta \mathrm{fh}_{i,t}^{\omega}, \Delta \mathrm{fc}_{i,t}^{\omega}\right\}_{i=1}^{N}$ ▷ Arrival of new information
    $s_t \leftarrow \mathcal{S}^W\left(\hat{s}_{t-1}, \omega_t\right)$ ▷ Pre-Decision update
    Compute maintenance check priorities for both check types

    procedure FIND THE OPTIMAL MAINTENANCE CHECK ACTION
      $X_t \leftarrow \{x_t | x_t = \mathscr{X}^{\pi}(s_t)\}$ ▷ Generate a set of feasible actions

      procedure DETERMINE THE BEST C- AND D-CHECK DECISIONS
        $g_k(\hat{s}_t, t+T) \leftarrow \sum_{i=1}^{N} \sum_{\tau=t}^{t+T} \hat{\eta}_{i,\tau}^k, \quad k \in K$
        $\overline{V}_t^{(1)}(s_t) \leftarrow$ Eq. (4.38)
        $\left\{x_{t,\mathrm{det}}^{\mathrm{A}*}, x_{t,\mathrm{det}}^{\mathrm{B}*}, x_{t,\mathrm{det}}^{\mathrm{C}*}, x_{t,\mathrm{det}}^{\mathrm{D}*}\right\} \leftarrow \mathrm{argmin}_{x_t \in X_t}\left\{C_t(s_t, x_t) + \gamma \overline{V}_t^{(1)}(s_t)\right\}$ ▷ $\hat{s}_t = \mathcal{S}^X(s_t, x_t)$
        $x_t^{\mathrm{C}*} \leftarrow x_{t,\mathrm{det}}^{\mathrm{C}*}, \quad x_t^{\mathrm{D}*} \leftarrow x_{t,\mathrm{det}}^{\mathrm{D}*}$ ▷ 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
        $g_k^{\omega}\left(\hat{s}_t, t+t_l, w_{t+1}^n\right) \leftarrow \sum_{i=1}^{N} \sum_{\tau=t}^{t+t_l} \eta_{i,\tau}^k\left(\omega_{\tau+1}^n\right)$ ▷ Simulation
        $G_k\left(\hat{s}_t, t+t_l\right) \leftarrow \frac{1}{n_{\mathrm{sample}}} \sum_{n=1}^{n_{\mathrm{sample}}} g_k^{\omega}\left(\hat{s}_t, t+t_l, w_{t+1}^n\right)$
        $\overline{V}_t^{(2)}(s_t) \leftarrow$ Eq. (4.45)
        $\left\{x_t^{\mathrm{A}*}, x_t^{\mathrm{B}*}, x_t^{\mathrm{C}*}, x_t^{\mathrm{D}*}\right\} \leftarrow \mathrm{argmin}_{\left\{x_t^{\mathrm{A}}, x_t^{\mathrm{B}}, x_t^{\mathrm{C}*}, x_{t,}^{\mathrm{D}*}\right\} \in X_t}\left\{C_t\left(s_t, \left\{x_t^{\mathrm{A}}, x_t^{\mathrm{B}}, x_t^{\mathrm{C}*}, x_{t,}^{\mathrm{D}*}\right\}\right) + \gamma\lambda \overline{V}_t^{(2)}(s_t)\right\}$
      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
      $\hat{s}_t^* \leftarrow \mathcal{S}^X(s_t, x_t^*)$ ▷ Post-Decision update
      $\hat{s}_t \leftarrow \hat{s}_t^*$
    end procedure

    $t \leftarrow t+1$
  end while

end procedure

4.5.7. ALGORITHM COMPLEXITY

To access the algorithm complexity of the lookahead ADP methodology, we count how many times the state transition function (4.7) 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 decisions for all check types. For comparison purpose, we also provide the algorithm complexity analysis for the *myopic* and *thrifty* policies.

Since there is only one state on a day t in stochastic AMCS, each state s_t has at most n_{act} actions (n_{act} depends on maintenance capacity), in *myopic* policy, we have to check all possible 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 (4.7) 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 letter checks). Instead of evaluating all possible actions on the day t, we check the hangar capacity first and see how many maintenance checks that the hangars can accommodate, then choose the action that fits the most maintenance checks in the hangars. Therefore, we just need to call (4.7) only once on the day t in the *thrifty* policy.

In the lookahead ADP methodology, it makes the aircraft maintenance check decisions in two steps. It first determines the optimal actions for aircraft C- and D-checks, and then for aircraft A- and B-checks. 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 (4.7) 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 (4.7) t_h times. Multiplying t_h with the number of actions n_{act} implies the algorithm complexity of the first step:

$$n_{\text{act}}\, t_h \tag{4.47}$$

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 (4.7) t_l times for each sample path. The total number of sample paths n_{sample} makes us call (4.7) $n_{\text{sample}}\, t_l$ 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, $n_{\text{act}}^{\text{A}}$, is smaller than n_{act}. The algorithm complexity of the second step is:

$$n_{\text{act}}\, n_{\text{sample}}\, t_l \tag{4.48}$$

Summing (4.47) and (4.48) gives the following algorithm complexity of determining the optimal action on a day t in the lookahead ADP methodology:

$$n_{\text{act}}\, t_h + n_{\text{act}}^{\text{A}}\, n_{\text{sample}}\, t_l < n_{\text{act}} \left(t_h + n_{\text{sample}}\, t_l \right) \tag{4.49}$$

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

Table 4.2: A- and C-check intervals of Airbus A319, A320 and A321 [28].

Aircraft Type	Check Type	Calendar Days	Flight Hours	Flight Cycles
A319	A-Check	120	750	750
	C-Check	730	7500	5000
A320	A-Check	120	750	750
	C-Check	730	7500	5000
A321-1	A-Check	120	750	750
	C-Check	730	7500	5000
A321-2	A-Check	120	750	750
	C-Check	1096	12000	8000

4

4.6. RESULTS

The proposed ADP methodology is evaluated using the aircraft maintenance data, and daily utilization from a European airline [27]. 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. Hence, there are only A- and C-check in the evaluation and Table 4.2 presents the associated inspection interval of each aircraft type. Two case studies are presented in this evaluation: the first case uses the data from the historical period September 25$^{\text{th}}$ 2017 to December 31$^{\text{st}}$ 2021 and has aircraft type A319, A320, and A321-1; the second case focuses on the period of March 20$^{\text{th}}$ 2019 to December 31$^{\text{st}}$ 2023 and has all four aircraft type. For each test case, there are five schedule/policies/methodologies tested:

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

M.2 The optimal deterministic AMCS schedule planned by [16], 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 (4.38)—(4.40) 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**.

4.6.1. MAINTENANCE ACTIONS

The airline has at most two A-check slots per workday and three C-check slots per day during the C-check period, but there are at least three days between the start dates of two successive C-checks. 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. 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 4.3.

Table 4.3: Possible aircraft maintenance check actions on a day t.

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

4.6.2. 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.

To validate the proposed lookahead ADP methodology, we use 100 test runs in 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 itself 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 [16].

Table 4.4: Model parameters for Stochastic AMCS optimization

Parameters	Description	Value	Unit
R_{lb}^{A}	A utilization threshold to prevent scheduling A-check too often [16]	21	day
R_{lb}^{C}	A utilization threshold to prevent scheduling C-check too often [16]	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

4.6.3. MODEL PARAMETERS

We assign 10^5 FH to λ to avoid creating an extra maintenance slot, based on the observation of objective values from our previous research. The objective values of the deterministic AMCS have a magnitude of 10^5 [16]. Setting $\lambda = 10^5$ can avoid creating unnecessary additional maintenance slots. We assign 10^{20} FH to ξ to penalize the action of wasting all available maintenance slots of a day when the lookahead policy predicts an extra maintenance slot needed 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 a decision, 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 anticipate the next check. A summary of model parameters is presented in Table 4.4.

4.6.4. OUTCOMES FOR THE TEST CASE 2017–2020

We first look at the KPIs of the test case 2017–2020. As shown in Table 4.5, 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, and, without looking into the future cost, the *myopic* policy is too greedy in A- and C-check scheduling. Although these two approaches at the moment of planning achieve higher aircraft utilization for both check types, the airline has to face extra costs to create additional maintenance capacity if one of the plans is executed.

Conversely, the *thrifty* policy does not need to create any extra maintenance slot for

Table 4.5: 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.

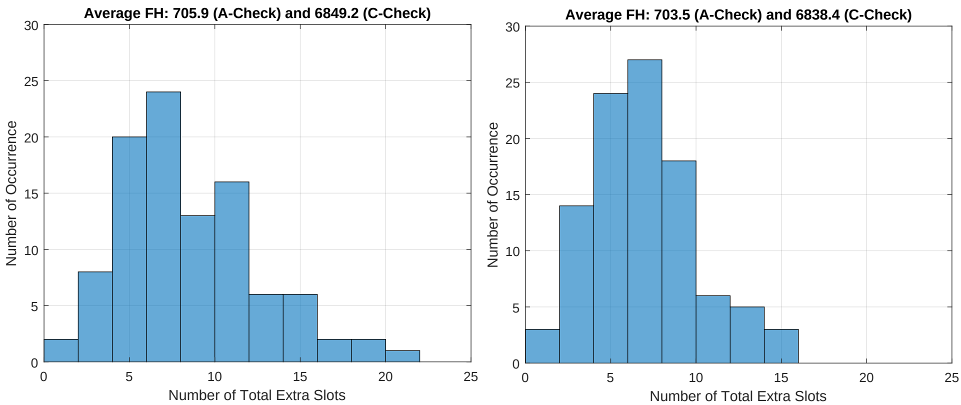
KPI 2017–2020 (1194 days)		Airline Schedule	Stochastic Results (100 test runs)				
			DP-based	Myopic	Thrifty	ADP-D	ADP-DS
C-check	Mean Average FH	6646.8	6785.4	7142.1	6200.6	6849.2	6838.4
	Mean Extra Slots	15	90.4	368.4	0.0	6.5	5.7
	Mean Total Checks	88	77.0	75.3	83.1	79.2	79.4
A-check	Mean Average 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
Computation Time/day [s]		—	0.02	0.09	0.05	0.35	2.63

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.

On the other hand, 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, as well as the *myopic* and *thrifty* policies.

The proposed lookahead ADP methodology combining deterministic with 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 4.3 shows the distributions of total extra slots under the ADP-D and ADP-DS for the 100 test 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 terms of fewer additional slots for both check types and of average aircraft utilization. Furthermore, Table 4.6 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.



(a) (b)

Figure 4.3: 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).

Table 4.6: 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

4.6.5. OUTCOMES FOR THE TEST CASE 2019–2022

Table 4.7 shows that the KPIs of the second test case follows a similar trend to the first test case. The *myopic* policy still 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 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 19.4/22.9 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, it associated mean average FH is the second-highest for both check type, only after the *myopic* policy.

On the other hand, 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 computation time in the ADP-DS 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 4.4, we are aware of the fact that ADP-DS uses

Table 4.7: 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 (1383 days)		**Airline Estimation**	**Stochastic Results (100 test runs)**				
			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	≥ 20.0*	22.9	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	≥ 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		40.0*	42.3	944.3	0.9	16.0	7.0
Computation Time/day [s]		—	0.02	0.09	0.05	0.35	2.63

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 4.8.

Table 4.8: 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	1.6×10^{-23}	99	6.8283

4.6.6. PRACTICAL 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. However, 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. Besides, 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 on a weekly basis.

Besides, 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 our case study, we assumed that an additional maintenance slot is very costly, more expensive than the cost of anticipating the maintenance check a few flight hours before the end

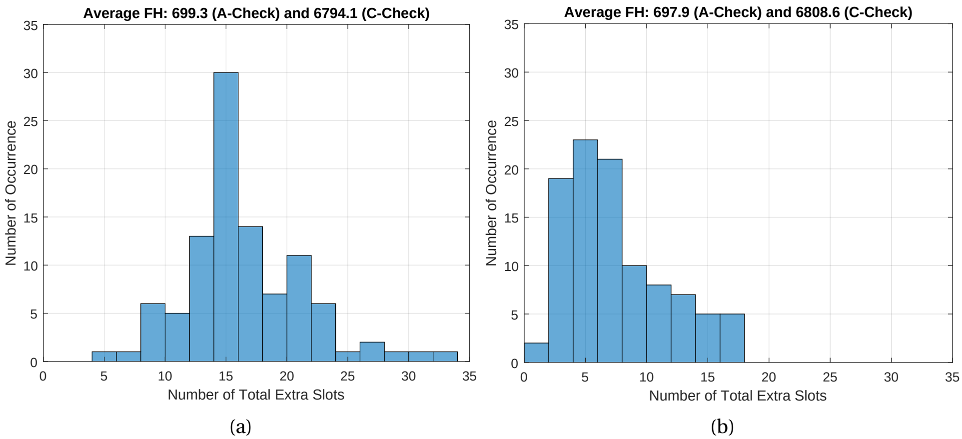


Figure 4.4: Distributions of total extra slots under two methodologies for the test case 2019–2022, under 100 test sample paths: (a) Distribution of total extra slots under the ADP-D (lookahead ADP using only deterministic forecasts); (b) Distribution of total extra slots under the ADP-DS (lookahead ADP using both deterministic and stochastic forecasts).

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 [16] and the lookahead ADP methodology itself using only deterministic forecasts.

4.6.7. SENSITIVITY ANALYSIS FOR 2019–2022

This subsection investigates the impact of model parameters of the lookahead ADP on the results of the stochastic 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 reduce the cost of generating an extra maintenance slot in the lookahead ADP methodology, how will that affect the solutions (KPIs)?

The baseline scenario is the ADP-DS from Table 4.7. 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 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 $n_{\text{sample}} = 20$ (240 in total for 12 actions) for the lookahead ADP methodology. For **Q.2**, if we increase the number of sample paths for Monte Carlo simulation, e.g., from 50 (600 in total for 12 actions) to 80 (960 in total for 12 actions), 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

Table 4.9: 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.

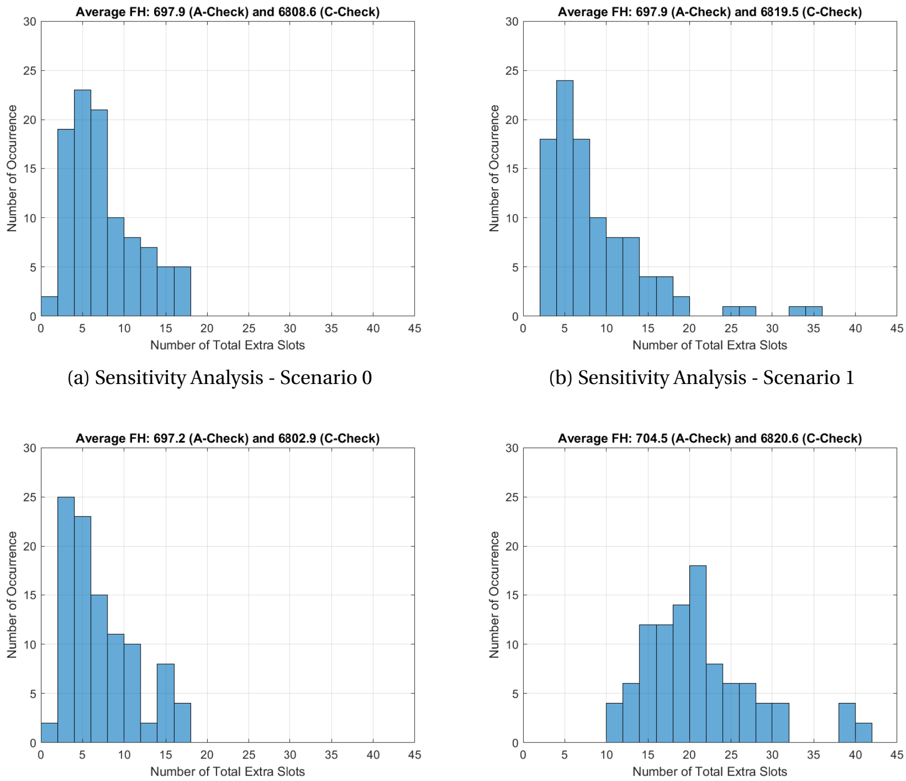
KPI of 100 Runs (2019-2022)		Scenario 0	Scenario 1	Scenario 2	Scenario 3
C-check	Mean Average FH	6808.6	6819.5	6798.0	6820.6
	Mean Extra Slot	4.0	4.9	3.8	5.4
	Mean Total Checks	90.6	90.7	90.8	90.4
A-check	Mean Average FH	697.9	697.9	697.4	704.5
	Mean Extra Slot	3.0	3.2	2.9	15.4
	Mean Total Checks	1019.9	1020.1	1020.6	1010.0
Mean Total Extra Slots		7.0	8.1	6.7	20.8
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

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 $\lambda = 100$. To investigate **Q.1**–**Q.3**, we set up the following test scenarios:

- *Scenario 0*: the baseline scenario, as pre-computed in the previous subsection;
- *Scenario 1*: conditions from *Scenario 0* and setting the number of random sample paths in the Monte Carlos simulation to 20, namely, $n_{\text{sample}} = 20$;
- *Scenario 2*: conditions from *Scenario 0* and setting the number of random sample paths in the Monte Carlos simulation to 80, namely, $n_{\text{sample}} = 80$;
- *Scenario 3*: conditions from *Scenario 0* and setting the penalty of creating one additional maintenance slot to 100 FH, namely, $\lambda = 100$;

We generate 100 test sample paths for each scenario and apply the lookahead ADP methodology to the stochastic AMCS. 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 4.5a and Figure 4.5b, 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 4.5c 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%.



(a) Sensitivity Analysis - Scenario 0

(b) Sensitivity Analysis - Scenario 1

(c) Sensitivity Analysis - Scenario 2

(d) Sensitivity Analysis - Scenario 2

Figure 4.5: 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 $\lambda = 100$.

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 4.5a and 4.5c). The A-check contributes to most of the extra slots. The approximation of cost function, $\overline{V}_t^{(1)}(s_t)$ in (4.38), 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, changing the cost of creating an extra slot λ only has a minor impact on C-check slot scheduling. On the other hand, we can perform multiple A-checks on a day; decreasing λ will inevitably increase the number of extra A-check slots (see Figure 4.5d). Consequently, there is more flexibility in performing aircraft A-check because of the creation of extra A-check slots; the number of merged A- in C-checks is reduced by 23.3%.

4.7. CONCLUSION

This chapter proposes a lookahead approximate dynamic programming (ADP) methodology to address the stochastic aircraft maintenance check scheduling (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 is capable of providing daily optimal maintenance check decisions and minimizing the total unused FH between checks. It increases aircraft availability and reduces the frequency of creating extra maintenance slots in the long term, and eventually 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. After that, based on the optimal C- and D-checks, 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.

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 creates additional maintenance slots to cope with the uncertainty from aircraft utilization and maintenance elapsed time. On the other hand, comparing KPIs from the maintenance schedule/estimation of the airline and 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 stochastic AMCS optimization. The lookahead ADP methodology can help maintenance planners react to changes in maintenance activities or tasks and promptly update the maintenance check actions. Maintenance planners can use the proposed methodology to update short-term schedules (for the following 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. For instance, to incorporate condition-based maintenance by considering the health prognostics and diagnostics and define 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 stochastic AMCS to the task level, producing an optimally integrated maintenance check and task execution plan.

REFERENCES

[1] Transport Canada, *Aeronautical Information Manual, LRA - 2.0 Aircraft Airworthiness, Airworthiness Directives,* (2008), (Accessed on April 24, 2020).

[2] Commercial Aviation Safety Team, *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 (2013), (Accessed on April 24, 2020).

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

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

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

[6] J. Brownlee, *What It Was Really Like To Fly During The Golden Age Of Travel,* (2013), (Accessed on September 27, 2019).

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

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

[9] J. M. Rosenberger, A. J. Schaefer, D. Goldsman, E. L. Johnson, A. J. Kleywegt, and G. L. Nemhauser, *SIMAIR: A STOCHASTIC MODEL OF AIRLINE OPERATIONS,* in *Proceedings of the 2000 Winter Simulation Conference* (2000).

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

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

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

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

[14] A. E. E. Eltoukhy, Z. X. Wang, and F. T. S. Chan, *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 (2018).

[15] C. Lagos, F. Delgado, and M. A. Klapp, *Dynamic Optimization for Airline Maintenance Operations,* Transportation Science **54**, 855 (2020).

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

[17] N. J. Boere, *Air Canada Saves with Aircraft Maintenance Scheduling,* Interfaces **7**, 1 (1977).

[18] M. Başdere and U. Bilge, *Operational aircraft maintenance routing problem with remaining time consideration,* European Journal of Operational Research **235**, 315 (2014).

[19] W. B. Powell, *Approximate Dynamic Programming - Solving the Curses of Dimensionality* (Wiley-Interscience, New York, 2011).

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

[21] J. S. McGrew, J. P. How, B. Williams, and N. Roy, *Air-Combat Strategy Using Approximate Dynamic Programming,* Journal of Guidance, Control, and Dynamics **33** (2010), https://doi.org/10.2514/1.46815.

[22] D. Zhang and D. Adelman, *An Approximate Dynamic Programming Approach to Network Revenue Management with Customer Choice,* Transportation Science **43** (2009), https://doi.org/10.1287/trsc.1090.0262.

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

[24] V. Schmid, *Solving the dynamic ambulance relocation and dispatching problemusing approximate dynamic programming,* European Journal of Operational Research **219**, 611 (2012).

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

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

[27] Q. Deng, *Data set for stochastic aircraft maintenance check scheduling*, https://doi.org/10.4121/12993437.v1 (2020), dataset.

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

5

A Decision Support System For Aircraft Maintenance Planning

This chapter presents a decision support system (DSS) for aircraft maintenance planning optimization. The DSS serves as a modeling framework that incorporates aircraft maintenance check scheduling (AMCS) optimization, optimal maintenance task allocation, and shift planning. It is capable of providing a comprehensive fleet maintenance plan for maintenance planners of airlines, including an optimal maintenance check schedule, task executions of each maintenance check, and the work shifts for the coming two weeks.

The content of this chapter is based on the following research article:

Deng, Q., Santos, B. F., and Verhagen, W. J. C. (2021). A Novel Decision Support System for Optimizing Aircraft Maintenance Check Schedule and Task Allocation. *Decision Support Systems.* To cite this article, please use the DOI https://doi.org/10.1016/j.dss.2021.113545.

5.1. PROBLEM DEFINITION

Aircraft maintenance is a sequence of activities, including overhaul, repair, inspection, or modification of an aircraft or aircraft systems, components, and structures to ensure an aircraft retains an airworthy condition. In the aviation industry, a commercial aircraft must undergo regular maintenance to prevent component and system failures during operations. Many of the aircraft maintenance activities take place after an aircraft has been operating certain flight hours (FH), flight cycles (FC), or calendar days (DY). The FH, FC, and DY are known as usage parameters to indicate aircraft utilization. The maximum usage parameters allowed in operation are defined as inspection intervals.

Modern aircraft have thousands of parts, systems, and components that need to be recurrently inspected, serviced, and replaced. Many airlines adopt a top-down approach to plan aircraft maintenance:

- **Step 1 – Maintenance Check Scheduling**
  First group major maintenance tasks with the same or similar inspection intervals into letter checks: A-, B-[1], C- and D-check, as showed in Table 5.1. Each check type is coupled with an elapsed time (time required for the execution of tasks within letter checks + time estimated for other tasks). Maintenance planners then create a maintenance check schedule (3–5 years for C-/D-check and 6–12 months for A-check) according to pre-defined elapsed time of each check type. The letter checks are performed in the hangar.

- **Step 2 – Maintenance Task Allocation**
  Although some tasks can quickly be packaged into these letter checks, a large number of other tasks (e.g., more than 70% for an Airbus A320 aircraft) are dephased from the inspection intervals of these checks. It means that they either have to be allocated to a more frequent letter check or manually allocated by maintenance operators to different maintenance events based on the suitability of the task to that check and the urgency of performing the task.

Table 5.1: Aircraft letter check and corresponding inspection interval [1].

Check Type	Interval	Maintenance Tasks
A-check	2-3 months	External visual inspection, filter replacement, etc.
B-check	—	Rarely mentioned in practice
C-check	18-24 months	Inspection of the individual systems and components
D-check	6-10 years	Inspection of most structurally significant items

Despite the rapid expansion of the global air travel industry and the increase of fleet size, the advances in aircraft maintenance planning (AMP) have been struggling to keep up with the times. In practice, AMP involves scheduling maintenance checks to each aircraft, allocating tasks to each check, planning the workforce for each task, inventory optimization, and coordination of maintenance tools. For small airlines, AMP is not so

[1] B-checks are rarely mentioned in practice. The tasks within B-checks are commonly incorporated into successive A-checks.

demanding and can be done manually according to the experience of maintenance planners. For large airlines, the AMP problem becomes more complex – maintenance planners have to spend several days or weeks on scheduling maintenance activities because of the lack of efficient tools. Since, on average, 9%—10% of the total cost of airlines goes to aircraft maintenance, which is equivalent to about $2.5M per aircraft per year [2], the savings derived from efficient AMP can be very substantial.

To facilitate the AMP process, many companies engage in developing AMP systems. For example, Ref. [3] developed one of the first AMP tools to improve maintenance efficiency and reduce associated cost. After that, many companies followed and developed various tools, e.g., Solumina MRO from iBASEt, Airline Suite from C.A.L.M Systems INC., WinAir from AV-Base Systems, and Maintenix from IFS, etc. To our best knowledge, all the available tools focus on managing and tracking the status of the maintenance tasks, providing a valuable computer-aid solution to manual planning. However, none of them has the function of producing an optimized maintenance schedule automatically.

AMP is challenging due to the lack of optimization approaches for planning maintenance checks and associated tasks, even though there are many available computer-aid solutions. Two distinct limitations in the current academic and industrial state of the art can be discerned, as further discussed in Section 5.2: 1) a lack of decision support system (DSS) to optimize the maintenance check (A-, B-, C- and D-checks) schedule; 2) a lack of DSS for optimizing aircraft maintenance check and task execution in an integrated manner. In the literature, there is no work integrating the two problems in a single optimization framework.

5

In 2015, the AIRMES project was launched by Clean Sky Joint Undertaking, a public-private partnership between the European Commission and the European aeronautics industry, to optimize end-to-end maintenance activities within an airline operator's environment [4]. We developed a DSS during the project to automate the maintenance planning process and provide maintenance check scheduling optimization, task allocation, and shift planning in one comprehensive solution. The contribution of our research is threefold:

- The DSS integrates aircraft maintenance check scheduling, maintenance task allocation, and work shift planning in the same framework. In practice, these processes are solved using different tools, while in the literature, these are seen as three different problems handled separately.
- We demonstrate that the DSS can improve aircraft utilization and reduce maintenance costs, compared with the current practice of airlines. It reduces the time needed for AMP from days or hours to 20–30 minutes.
- We also present the usefulness of the DSS in helping airlines evaluate different aircraft maintenance strategies before implementation.

This paper presents the architecture of the resulting DSS and the corresponding optimization modules for maintenance check schedule, task allocation, and shift planning. We also discuss the applicability of the DSS by presenting the results from a case study with a European airline and several industry partners. The case study validates the utility of the DSS for both maintenance planning optimization and future scenario analysis.

The outline of this paper is as follows: Section 5.2 gives an overview of the relevant literature on the aircraft maintenance domain. The DSS architecture is presented in Section 5.3, including aircraft maintenance check scheduling optimization, task allocation, and shift planning as well as their corresponding algorithms. Section 5.4 describes the demonstration exercise with data from the partner airline. The last section summarizes the research with concluding remarks and gives an outlook on future work.

5.2. Related Work

The aviation industry is extremely competitive in Europe. The average net profit of airlines usually represents only up to 4%–5% of revenues and about 9%—10% of the total cost goes to aircraft maintenance [2]. Efficient AMP is one useful way of reducing maintenance costs. The benefit of efficient AMP is two-fold: on the one hand, the increased aircraft availability indicates that there will be more aircraft available for commercial operations, and eventually, generating more revenues; on the other hand, it decreases the number of aircraft maintenance inspections, and therefore, reduces the maintenance operation costs in the long term. This section reviews the previous research on AMP from long-term planning (3–5 years) to short-term planning (several days to weeks).

5.2.1. Long-Term Aircraft Maintenance Planning

Long-term AMP aims to generate an aircraft heavy maintenance schedule (C- and D-checks) before determining the tasks within each check, also known as aircraft maintenance check scheduling (AMCS). It is indispensable since C-check has an interval of 18–24 months, and D-check is usually scheduled once every 6 years; airlines need a C- and D-check schedule to further plan the A- and B-checks and the associated tasks for all the (A-, B-, C-, and D-) checks. In 1977, Air Canada developed one of the first DSSs for the long-term AMCS, called AMOS [3]. AMOS was considered a computer-aid manual planning approach since the developers did not see the value of finding an optimal solution that could rapidly become obsolete due to uncertainty. It helped Air Canada reduce the time for planning a 5-year C-check schedule for its fleet from 3 weeks to a few hours. Besides, Ref. [3] defined the long-term (3–5 years) planning, and it is the only available reference of the long-term AMP category before 2020.

Following this research direction, Ref. [5] proposed a dynamic programming (DP) based methodology for long-term AMCS within the AIRMES project in 2020, adopting the assumptions, problem formulation presented in [3]. It aimed to optimize the aircraft maintenance check schedule for the future 3–5 years. This work is the first step towards building an integrated AMP framework, focusing on long-term AMP. The DP-based methodology generates an optimized 4-year schedule for both light and heavy maintenance within 15 minutes. The optimized maintenance check schedule can be further used to plan the maintenance tasks within each check and daily work shift.

5.2.2. Short-Term Aircraft Maintenance Planning

In contrast to the little available literature about long-term AMCS, there are many studies on short-term AMP in the topics of maintenance routing, maintenance personnel management, and maintenance task scheduling. The reason is that by optimizing short-term maintenance activities, airlines can see tangible benefits in a few days or weeks.

Aircraft Maintenance Routing

Aircraft maintenance routing (AMR) is to design flight routes for every aircraft to meet the maintenance requirements set by Federal Aviation Administration (FAA) and individual airline companies. Extensive research works have contributed to AMR through flight schedule design [6, 7], determining routes flown by each aircraft [8–10], fleet assignment (assigning an aircraft model for each flight) [11–13], tail assignment (determining which aircraft should fly which segment) [14–16], or even addressing the aircraft routing in conjunction with crew pairing [17–19]. These studies usually consider aircraft maintenance as an operational requirement but did not plan the maintenance checks or tasks.

Maintenance Personnel Planning

Maintenance personnel planning (MPP) is one of the main research directions of short-term AMP. An effective maintenance workforce supply can reduce operations costs while ensuring aviation safety and punctuality. It has attracted lots of attention from both industry and academia. Early in 1994, KLM Royal Dutch Airline and Erasmus University Rotterdam developed a DSS to smooth the workload of aircraft maintenance personnel by increasing the number of peaks of workloads and reducing the peak length [20]. It helped KLM improved the utilization of maintenance technicians (the ratio of productivity labor-hours to total available labor-hour). After that, many researchers envisioned the potential benefits and continued the MPP study, such as optimizing the workforce supply [21–23], or minimizing the total labor cost [24, 25]. However, MPP usually assumes that maintenance tasks are given rather than planning the tasks.

Maintenance Task Scheduling

Maintenance task scheduling (MTS) refers to allocating maintenance tasks to time slots so that the tasks can be executed before due dates. It includes task scheduling for aircraft line maintenance (coordinating maintenance tasks to be carried out at the gate during turnaround time and the required maintenance resources), daily hangar maintenance, or work shift. There are some studies addressing the MTS for line maintenance, such as spreading the workload more uniformly across shifts [26], improving aircraft availability and reducing maintenance costs [27], or optimizing both workforce and tasks [28]. MTS for line maintenance planning has an operational nature. It only focuses on optimizing a limited number of maintenance tasks during aircraft turnaround time.

Task scheduling for daily aircraft hangar maintenance can be seen in [29]. According to the authors, optimizing the daily hangar maintenance tasks to be executed 24 hours beforehand also maximizes the availability of fighting jets for the missions of the next day. The authors call attention to the fact that if we want to plan the daily maintenance task for each letter check, we have to look into a planning horizon longer than 24 hours, especially for the C-/D-check. Besides, the daily maintenance task plan bridges the gap between AMCS and associated work shift planning. That is, we can better plan each morning/afternoon/evening shift and prepare the tools and aircraft spare parts if we know the daily maintenance tasks in advance. Hence, Ref. [30] proposed a bin packing approach to determine daily maintenance tasks (for each A-/B-/C-/D-check) given a long-term (3–5 years) maintenance check schedule for AIRMES. As a result, it gives a long-term (3–5 years) plan of maintenance tasks for each day and a heterogeneous fleet.

5.2.3. Concluding Remarks for Literature Review

To our best knowledge, most of the studies in the AMP domain focus either on AMR or MPP, assuming that the maintenance tasks are given. There are some studies on MTS, yet most of them focus on line maintenance problems. The long-term and short-term AMP was not yet considered in a single framework, nor was a DSS presented in the literature addressing the AMP. Synthesizing the literature review gives rise to two challenges in the AMP domain:

1. No DSS for aircraft maintenance planning optimization (AMPO) is presented in the academic literature that can generate an optimally integrated maintenance check and task execution plan at the fleet level.

2. Commercial DSSs addressing the fleet maintenance check level are relatively rare. Even so, they do not optimize the maintenance check schedule.

5

In practice, maintenance planners have to spend a significant amount of time and effort scheduling the aircraft letter checks and coordinating associated tasks execution activities. It can happen that with the aid of current DSSs, the maintenance planners still obtain an inefficient plan; this may, in the long-term, result in more letter checks and higher operation costs.

The DSS presented in this paper contributes to bridging two main research streams, long-term and short-term AMP, by integrating the AMCS problem and its methodology presented in [5], the MTS problem and the associated algorithm presented in [30], and a shift planning approach into the same framework.

5.3. System Architecture

To address the challenges identified in Section 5.2, we developed a DSS specifically for AMP using the programming language Python and for Windows operating system. The DSS is a stand-alone software prototype and has already been converted to an executable file. It can be run on any individual PC without installation or a license. The DSS consists of three components (layers), a database, a model, and a graphical user interface (GUI):

- *Database*: Store the input data, including the maintenance planning document (MPD) for aircraft manufacturers, fleet status, operational constraints, and available workforce from airlines.

- *Model*: Clean and process input data, optimize the aircraft maintenance check schedule and maintenance task execution plan.

- *Graphical User Interface* (GUI): Allow users to interact with the DSS and visualize the planning results and the associated KPIs.

In this section, we present the structure of the DSS layer by layer, as illustrated in Figure 5.1. We begin with description of database layer (Subsection 5.3.1) and input, followed by a detailed introduction of the optimization models and algorithm (Subsection 5.3.2). In subsection 5.3.3, we outline the GUI of the DSS.

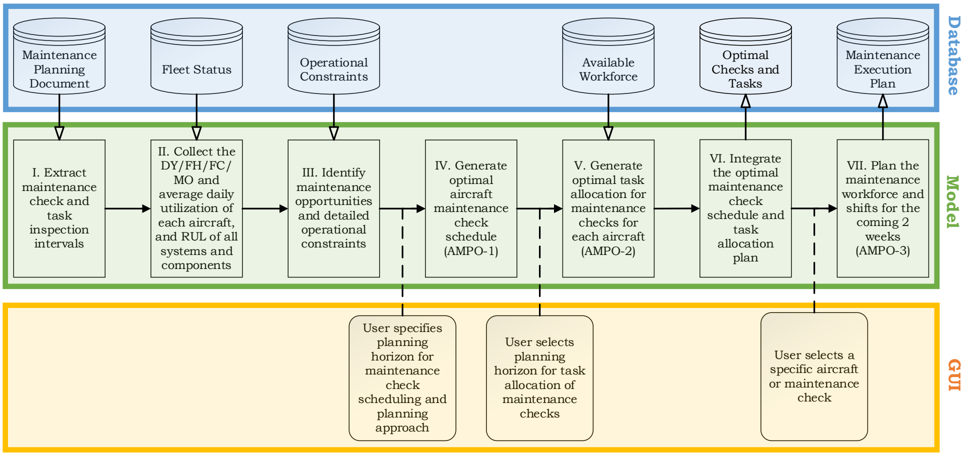


Figure 5.1: Architecture of the decision support system for AMPO.

5.3.1. DATABASE AND INPUT DATA

The database stores the input in the format of comma-separated values (CSV) and output in Excel. We classify the input into four categories:

MAINTENANCE PLANNING DOCUMENT

The maintenance planning document (MPD) is provided by the aircraft manufacturer. It specifies the maintenance tasks according to the aircraft structure, systems, and components, as well as corresponding inspection intervals (as described in Table 5.1). The MPD gives strict criteria for aircraft maintenance – all letter checks and tasks have to be performed before the corresponding usage parameters reached their maximums (intervals). Any violation of maintenance task execution will prevent the aircraft from flying because of safety concerns.

Table 5.2: An example of fleet status with respect to aircraft C-Check on 01/07/2020.

Fleet	Tail	Before	Next	DY	FH	FC	fh/day	fc/day	Phase-In
A320	AC-1	C 12.1	-1	212	2391	963	10.3	4.2	12/01/1998
A319	AC-2	C 10.1	C 11.1	607	6439	2600	9.9	4.1	08/06/1998
A321	AC-3	—	C 1.1	0	0	0	10.1	4.2	01/03/2021

FLEET STATUS

We use Table 5.2 to illustrate the structure input data. The column **Fleet** shows the aircraft type. **Tail No.** indicates the aircraft tail number. **Before** and **Next** represent the previous and next letter checks respectively. **DY**, **FH** and **FC** are the usage parameters of the fleet. **fh/day** and **fc/day** are the average daily utilization of the fleet. **Phase-In** indicates when an aircraft starts in commercial operation. This is relevant information as old aircraft will phase out after a certain number of checks, and meanwhile, airlines have new aircraft in operation. If an aircraft will phase out, we give "-1" to its next A-/C-/D-check, meaning that no more A-/C-/D-check needs to be scheduled. If the phase-in date of an aircraft is later than the current date, this aircraft only starts flying from the phase-in date, and before that, its usage parameters remain 0.

Operational Constraints

The operational constraints can be divided into two categories: commercial constraints and maintenance constraints. The operations center of airlines defines the commercial constraints. For example, the operations center may limit the availability of the aircraft to perform maintenance during commercial peak seasons (e.g., during the summer or specific holidays), or it may impose an earlier time limit to the maintenance check of a specific aircraft following the end of a leasing contract or the chartering of an aircraft to third parties.

The maintenance constraints are defined by the maintenance department, which specifies the maintenance capacity according to available maintenance resources, e.g., maintenance tools, workforce, and aircraft spare parts. This capacity is expressed as maintenance slots per day that define how many aircraft can be at the hangar for a specific type of maintenance. Furthermore, other maintenance constraints may apply, such as that no heavy checks can start on the same day to avoid high demanding works in parallel or that some aircraft already have maintenance predefined before computing the schedule. The latter takes place, e.g., when part of the maintenance program is executed by third-parties or partially depends on third-parties, not being subject to rescheduling. A typical example of this is the replacement of landing gears or engines. Besides, maintenance task execution follows the sequence of opening the access panel, inspection, maintenance, and closing the access panel.

Workload of Each Task and Available Workforce

The workload of each task is provided by the airline. Each task associates a task code, a set of skill types required to perform the task, labor hours for each skill type defined by the MPD. If there are urgent unscheduled tasks, they can be added to the input with corresponding duration, workforce, and due dates.

The available workforce is the input given by airlines and divided per skill types (e.g., engines and flight control systems, avionics, aircraft metallic structure, and painting, etc.). The available workforce includes the total number of maintenance technicians per skill type, the number of hours a technician work per day on average, and the number of available technicians on each week in the year. The available workforce constrains the task allocation to maintenance checks because it is limited per day, according to the daily workforce schedule. Since aircraft maintenance work is usually ongoing 24 hours every day, airlines divide the daily workforce into three groups of workers to perform their duties and call those groups *morning shift*, *afternoon shift*, and *night shift*. In the input data, the maintenance planners of airlines have to specify the maximum number of technicians in one shift and also for one task.

5.3.2. Optimization Model and Algorithms

The model layer has three optimization models in total: a maintenance check scheduling model (AMPO-1 in Figure 5.1), a maintenance task allocation model (AMPO-2), and a shift planning model (AMPO-3). The design of the model layer follows the top-down approach. The DSS first generates an optimal aircraft maintenance check schedule in AMPO-1, then allocates the maintenance tasks to each maintenance check in AMPO-2. After that, it plans the shifts according to the maintenance tasks to be executed in each letter check.

The reason for following the top-down approach is that it is impossible to plan the work shifts before knowing the task execution or plan all maintenance tasks one after another for the entire fleet without knowing the maintenance check schedule. The maintenance check schedule indicates in which letter check a maintenance task could be allocated without violating the safety regulation defined by the MPD. The work shifts can only be planned based on the maintenance check schedule and the tasks to be executed within each check. The overall optimization process entails the following seven steps:

Step I: Extract maintenance check and task inspection interval

The *Model* component extracts the maintenance check (A-/C-/D-check) interval and inspection intervals of all tasks from the MPD stored in the database. The inspection intervals are expressed in the form of max DY/FH/FC allowed in commercial operation.

Step II: Collect DY/FH/FC of each aircraft and remaining utilization of all systems and components

The *Model* component loads the fleet status (current DY/FH/FC for each check type since its previous execution) and average aircraft daily utilization (FH/day and FC/day) stored in the database. This process can be seen in the second step in the model layer of Figure 5.1. The *Model* component also collects the usage parameters of all aircraft systems and components and computes the remaining utilization of each system and component. For example, consider a component of an aircraft with max usage parameters 120 DY, 1000 FH, and 600 FC, and this aircraft has daily utilization of 10 FH/day and 5 FC/day. Given current usage parameters 500 FH and 250 FC, the remaining utilization of this component would be 50 days.

Step III: Identify maintenance opportunities and detailed operational constraints

According to the input constraints from the operation center and maintenance department of airlines, the *Model* component identifies the maintenance opportunities. The maintenance opportunities indicate the time-window when a specific check type is allowed to be performed and the corresponding check capacity. Table 5.3 presents a format of maintenance opportunities stored in the database after input processing:

For a specific maintenance check type, if a time window is not within any *Start Date* and *End Date* in Table 5.3, it means that the associated capacity for this period is 0.

Step IV: Generate optimal aircraft maintenance check schedule (AMPO-1)

After processing and loading the input data, the user can specify the planning horizon for aircraft maintenance check scheduling (AMCS) optimization. The default planning

Table 5.3: An example of maintenance opportunities stored in database.

Fleet	Check Type	Start Date	End Date	Capacity
A320	C-/D-Check	Oct-1-2017	May-31-2018	3
A320	C-/D-Check	Jun-1-2018	Jun-14-2018	1
A320	A-Check	Every Monday	Every Friday	1
A320	A-Check	Sep-26-2017	Sep-26-2017	2
⋮	⋮	⋮	⋮	⋮

horizon is three years to ensure that it includes at least one C-check for each aircraft, but the user can choose from two to six years.

The model formulation of AMPO-1 can be seen in Chapter 2. Currently, there is only one objective function within the DSS for AMCS optimization, minimizing the unused flight hours of the entire fleet [3] for a period specified by the user. It is possible to add more objectives or even multi-objectives later on. The optimal letter check schedule is generated using a dynamic programming (DP) based methodology, as presented in [5]. The idea is to check whether the maintenance capacity in the future is sufficient or not for each maintenance action (e.g., performing a C-check or several A-checks). This methodology follows a forward induction approach, incorporating a maintenance priority solution to deal with the multi-dimensional action vector, as well as a discretization and state aggregation strategy to reduce outcome space at each time stage. If the input data does not lead to a feasible maintenance check schedule, the DSS will suggest the best dates for adding necessary maintenance slots to make it feasible.

STEP V: GENERATE OPTIMAL TASK ALLOCATION FOR MAINTENANCE CHECKS FOR EACH AIRCRAFT (AMPO-2)

Once the AMPO-1 plans the optimal letter check schedule for the entire fleet, the DSS allocates the maintenance tasks to each letter check, assuming that there are sufficient aircraft spare parts and maintenance tools. The task allocation aims at minimizing the total cost in task execution, subject to the daily available workforce. It adopts an algorithm based on the worst-fit decreasing (WFD) [30]. The task allocation algorithm treats the maintenance resources within each check as bins and the maintenance tasks as items. It consists of:

- *Bin Definition*: The task allocation within AMPO-2 divides the entire aircraft letter check schedule into time segments (bins) according to the number of parallel maintenance checks. For example, in Figure 5.2, C1.2, C12.1, C7.1, C7.2, and C9.1 are the maintenance checks. T1–T7 are the bins defined by the AMPO-2. The sizes of the bins (time segments) are determined based on the aircraft maintenance resources, i.e., the number of maintenance technicians working during the time periods of the bins (the available workforce per day is given in the input).

- *Bin Selection*: The heuristic algorithm sorts the time segments according to the associated capacity (maintenance resources), from highest to lowest. When the algorithm selects a bin to allocate a maintenance task, it always starts with the

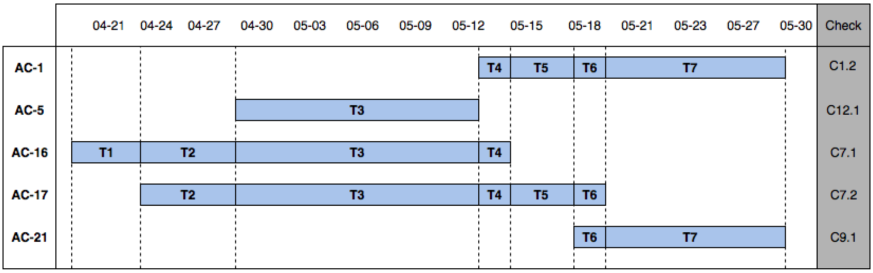


Figure 5.2: Overlapping maintenance checks are divided into several time segments (bins) in AMPO-2 - i.e., T1, T2, ..., T7.

bin with the highest remaining capacity. The availability of bin (time-segment) depends on the aircraft hat having letter check during that time-segment. In the example of Figure 5.2, T1 is only available for aircraft (AC) 16, T2 is available for both AC-16 and AC-17, etc.

- *Item Allocation*: The algorithm allocates the items (tasks) following the rules of "the most urgent item (task) first". Each maintenance task must be allocated before its due date; otherwise, it generates extra capacities and notifies the DSS user.

The model formulation and the associated task allocation algorithm of AMPO-2 are presented in Chapter 3. For the maintenance tasks that have to be executed in a strict order, the task allocation algorithm groups those tasks into a package. A task package is also considered one item. After that, the algorithm allocates the item (task package) to a bin (time segment of a maintenance check). In this way, it ensures that all tasks within the package will all be executed. For instance, the maintenance tasks presented in Table 5.4 have to be executed in the order of:

$$\text{1200-A} \longrightarrow \text{1200-B} \longrightarrow \text{1200-C} \longrightarrow \text{1200-D} \tag{5.1}$$

Table 5.4: An example of maintenance tasks that have to be executed in the order of A → B → C → D.

Fleet	Tail No.	Date	Item	Description
A320	AC-1	Mar-19-2019	1200-A	Open the panel at aircraft component xxx
A320	AC-1	Mar-19-2019	1200-B	Inspect aircraft component xxx
A320	AC-1	Mar-19-2019	1200-C	Replace component xxx
A320	AC-1	Mar-19-2019	1200-D	Close the panel at aircraft component xxx

In this example, technicians have to execute task 1200-A (open the panel at component xxx) first. Otherwise, they cannot continue to inspect or replace the component xxx. After the technicians complete the task 1200-C, they have to close the associated panel (close the panel at component xxx). The task allocation algorithm groups these four tasks into one package and label it as "Item 1200", providing information of the sequence when presenting the results to the user.

Step VI: Integrate the optimal maintenance check schedule and task allocation plan

In this step, the DSS first creates a folder for each aircraft with the name "aircraft tail number + Time + Date", and decouples the entire maintenance check schedule obtained from AMPO-1 according to aircraft tail numbers. In each folder, it saves the associated maintenance checks in the format of Excel. Next, the DSS organizes all the maintenance tasks from AMPO-2 within the same letter check in one table in CSV format and puts this CSV file in the folder according to the aircraft tail number of the letter check. The user can compare or keep track of the historical optimization results according to the time and date in the folder name.

Step VII: Plan the maintenance workforce and shifts (AMPO-3)

The *Model* component also has an algorithm (AMPO-3) to plan the maintenance work shift (morning/afternoon/night), create job cards, and estimate the workload after the AMPO-2 completes the task allocation for all letter checks. Due to the uncertainty associated with the workforce available per shift, the optimal maintenance check schedule and task execution plan may quickly become obsolete. Thus, the AMPO-3 only creates the work shifts and job cards for the initial 1–2 weeks of the planning horizon.

5

The shift planning algorithm allocates the maintenance tasks to each shift, respecting the workforce available per shift and the sequence of opening access panel, inspection, maintenance, and closing access panel (and this is the only task execution sequence we have to follow in both AMPO-2 and AMPO-3 according to the specification of our airline partner). Figure 5.3 illustrates the workflow of shift planning function (AMPO-3). AMPO-3 first assigns the tasks of opening the access panel to the morning shift. If there is no available workforce left in the morning shift, it continues to assign those tasks to the afternoon shift (or even night shift) until all the tasks of opening access panels are allocated. Next, the algorithm assigns the inspection works, and after that, the maintenance tasks. The tasks of closing the access panel are allocated at last. The shift planning process continues until it loops over the task execution plans of all maintenance checks. When it finishes, the DSS will store the results in the database according to the aircraft tail number of the tasks.

5.3.3. Graphical User Interface

The GUI serves the purpose of interacting with DSS users. The DSS users can load input data, start the AMPO, visualize the optimization results and associated KPIs, change operational constraints (planning horizon, the number of maintenance slots, or reserve slots for specific maintenance activities), and export the output data via the GUI. Those actions are the basic requirements for the GUI from the DSS users, identified by the AIRMES project group.

The GUI of the DSS has a single main window, divided into five screens. The user can see the maintenance check schedule of all aircraft for the entire planning horizon on a daily basis or per hangar view on different screens. The GUI also displays key performance indicators (KPIs), the tasks allocated per maintenance check, the workforce assigned per day (of the first few weeks), the identification of the maintenance interval tolerances used, the maintenance slots generated as additional to the given capacity. The

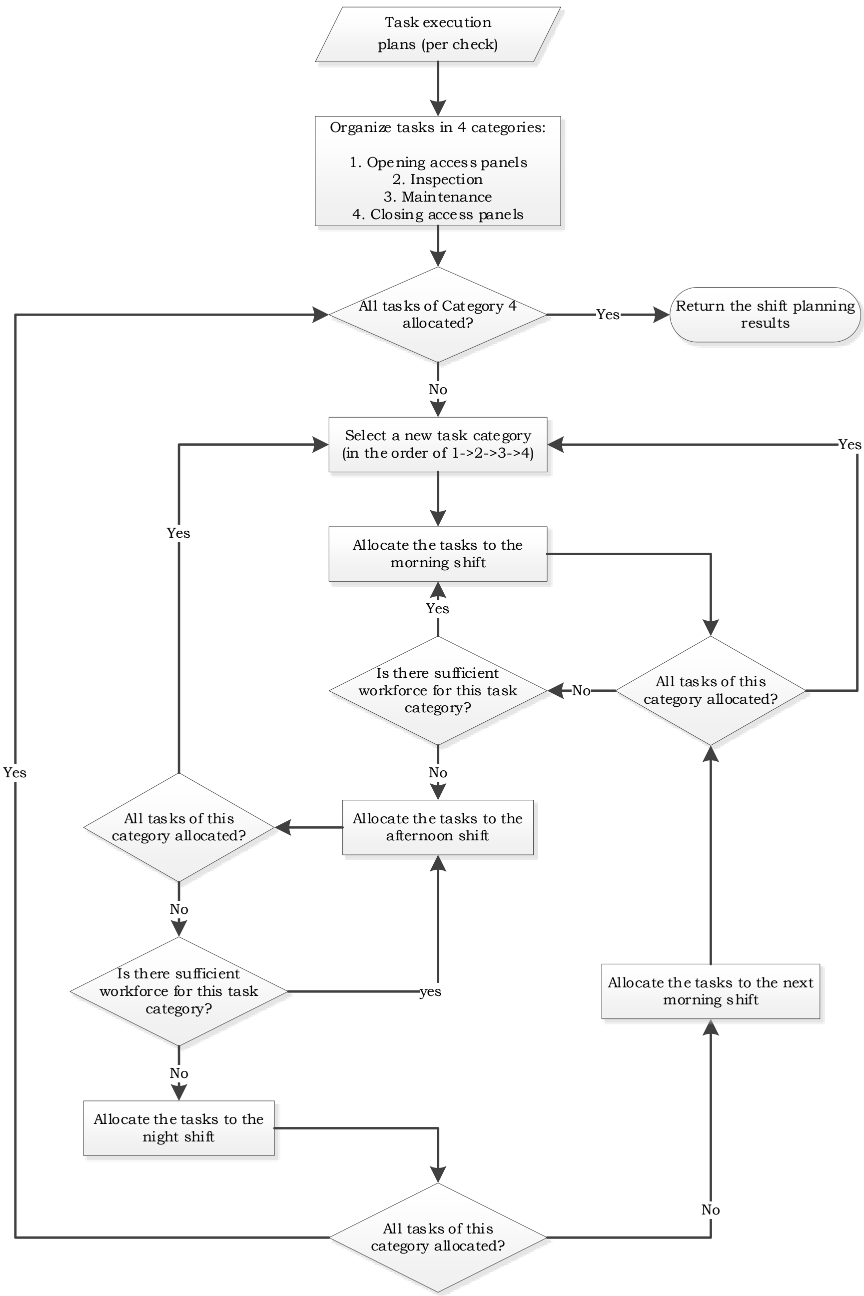


Figure 5.3: Workflow of shift planning (AMPO-3).

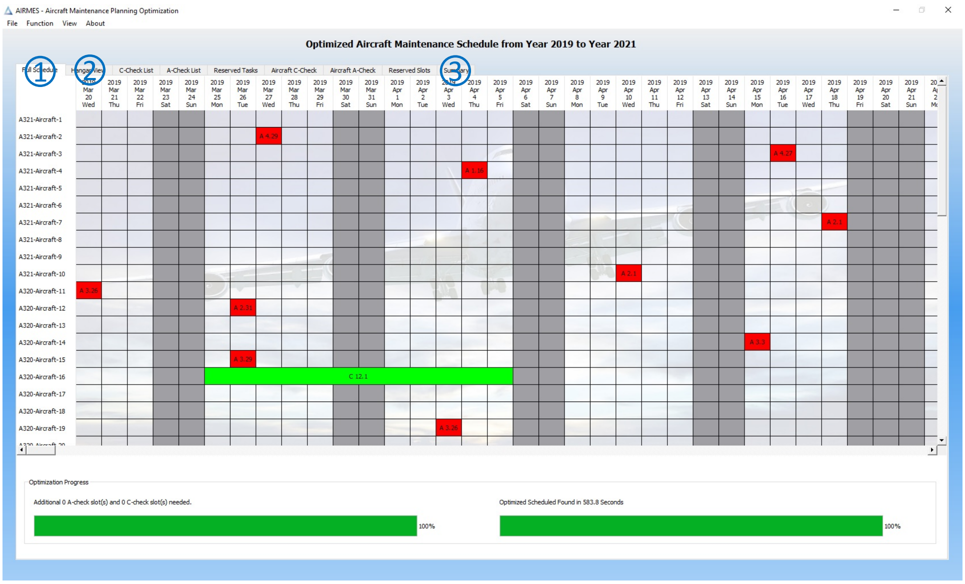


Figure 5.4: Main screen of the DSS.

user can also use the GUI to set the planning horizon, modify the start dates of specific maintenance checks, and change the operational constraints, such as adding/reducing maintenance slots or reserve maintenance slots for other maintenance activities. We use Figure 5.4 and 5.5 to illustrate the main features of the DSS.

In Figure 5.4, the marker (1) indicates the main screen of the DSS. The main screen displays the aircraft maintenance check schedule per day per aircraft, computation time, and the number of extra maintenance slots created during optimization for a specific planning horizon. The marker (2) indicates the 2$^{\text{nd}}$ screen of the DSS. The 2$^{\text{nd}}$ screen displays the maintenance check schedule for the entire fleet in the hangar view. The marker (3) indicates the screen of displaying the KPIs, including the mean FH, mean FC, total maintenance checks, distribution of unused FH and FC for each check type, and the number of merged A- in C-/D-Checks.

In the 2$^{\text{nd}}$ screen, the DSS user can further see the maintenance tasks of each check. If the DSS user selects a maintenance check, a dialogue box will be popped up to display the aircraft tail number, maintenance check label, current DY, FH, and FC. The user can click the button “Show Tasks”, as indicated by marker (4) in Figure 5.5. The DSS will display a list of maintenance tasks within the check and a figure that shows the workload distribution and the work shifts. The user can also change the start date of a specific check by clicking the button indicated by marker (5). The DSS will re-optimize the entire schedule according to the new specification from the user.

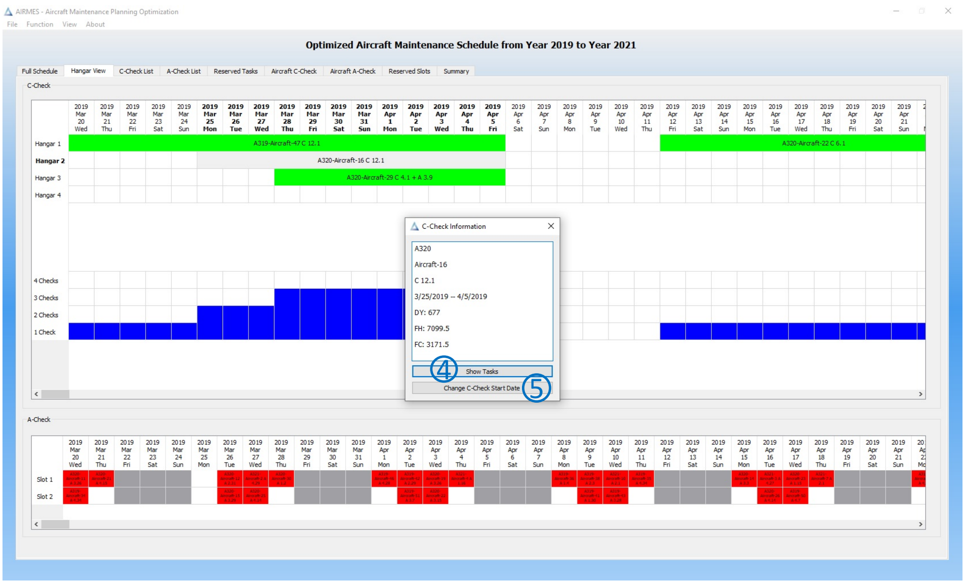


Figure 5.5: The 2nd screen of the DSS.

5.4. Demonstration and Evaluation

The DSS was demonstrated and validated in a demonstration exercise organized as part of the AIRMES project, on 51 aircraft, in March 2019 [31, 32]. The exercise was carried out in collaboration with one of the major European airlines and one of the leading aircraft manufacturers and observed by the Clean Sky 2 Joint Undertaking partners involved in the related research effort.

This exercise aimed to validate the value of the DSS and demonstrate that it can be implemented in practice, primarily for AMP optimization and the study of future maintenance scenarios. For this reason, two test cases were performed and discussed. In the first test case, we aimed to validate the DSS and benchmark its performance by comparing the solution obtained with the maintenance schedule of the airline. In the second test case, we investigated the current considerations of the airline about its future maintenance policies and fleet developments. The results were checked and validated by the airline experts involved in analyzing such maintenance policies, providing valuable insights to the airline on future maintenance limitations and solutions.

5.4.1. Standard AMCS Optimization on Fleet Maintenance Data

We received the input for AMCS on March 19th 2019 and optimized the A- and C-checks for the A320 family of our airline partner from March 20th 2019 to December 31st 2021, under the same operational constraints as the airline. According to the requirements of our airline partner, D-checks are merged within C-check in the following pattern:

$$\text{C-1},\quad \text{C-2},\quad \underbrace{\text{C-3}}_{\text{D-check}},\quad \text{C-4},\quad \text{C-5},\quad \underbrace{\text{C-6}}_{\text{D-check}},\text{C-7},\quad \text{C-8},\quad \underbrace{\text{C-9}}_{\text{D-check}},\ldots \tag{5.2}$$

We compared our results with the maintenance schedule available at the airline (Airline Schedule). According to the results illustrated in Figure 5.6 and 5.7, the AMPO-1 of the DSS outperforms the planning approach of the airline. The AMPO-1 results in 6946.5 FH for C-check and 705.1 FH for A-check, higher than 6783.8 FH and 701.1 FH from the maintenance schedule of the airline, but the result of AMPO-1 has one fewer C-check and three fewer A-checks. Our airline partner also checks the maintenance check schedule obtained using the DSS and agrees that the DSS generates a better schedule than the maintenance planners. Besides, the AMPO-1 of the DSS optimizes both the aircraft A- and C-check schedule for 2019–2021 within only 10 minutes. It means that the DSS user can run the DSS to update its aircraft maintenance check schedule if there are changes instead of manually shuffling the A-/C-checks to make another feasible one.

From a saving and revenue management perspective, since airlines spend on average \$150K–\$350K on a C-check [1] and \$10K–\$15K on an A-check, one fewer C-check and three fewer A-checks in total can result in a potential saving of \$0.1M–\$0.4M for the considered time horizon of roughly three years. Furthermore, a C-check lasts about 1–4 weeks, and an A-check lasts 24 hours in this case study. One reduced C-checks and three fewer A-checks are equivalent to about 10–31 days of aircraft availability for commercial operations. This may generate a considerable amount of revenue for the airline. According to an economic evaluation performed by another Clean Sky project, the implementation of the DSS can potentially reduce the base maintenance costs by 2.7% for point-to-point carrier airlines, 0.5% for large hub and spoke carrier airlines, and 2.4% for small hub and spoke carrier airlines [33].

Following the demonstration of AMPO-1, we continued to test the task allocation (AMPO-2) of the DSS. The AMPO-2 allocated the maintenance tasks using the optimal maintenance check schedule created by the AMPO-1. It addressed the task allocation for the maintenance check schedule of Mar 20th 2019–Dec 31st 2021 within 10 minutes. The outcome of AMPO-2 is an optimized task allocation plan for the entire fleet and all letter checks, including over 60,000 tasks. An example of the outcome from AMPO-2 can be seen in Figure 5.8. To verify the AMPO-2, we compared the optimal task allocation plan with the results from a commercial optimization solver. The comparison showed that the AMPO-2 produces results within an optimality gap of only 0.028%. Our airline partner also validated the optimal task allocation plan and its feasibility by benchmarking our solution with the task allocation solution they had for the following year. The maintenance planners of the airline stated that the results from optimal task allocation are feasible for practical implementation.

After the AMPO-2 completed the optimal task allocation for all maintenance checks, the AMPO-3 planned the work shifts and creates job cards for technicians. Our airline partner set the horizon of shift planning for two weeks. Table 5.5 shows an example of the results from AMPO-3. The 1st column of the table shows the aircraft tail number. The 2nd and 3rd columns indicate the date and work shifts. The 4th column describes the item or action, and the 5th column tells the maintenance planner where the maintenance work is in the aircraft. The last eight columns imply the workload needed for each skill type. The airline evaluated the work shifts after the demonstration and indicated that the work shifts of the first 2–3 days are almost the same as they planned, yet the difference becomes dramatic in the second week. It is worth mentioning that, for example, if a task

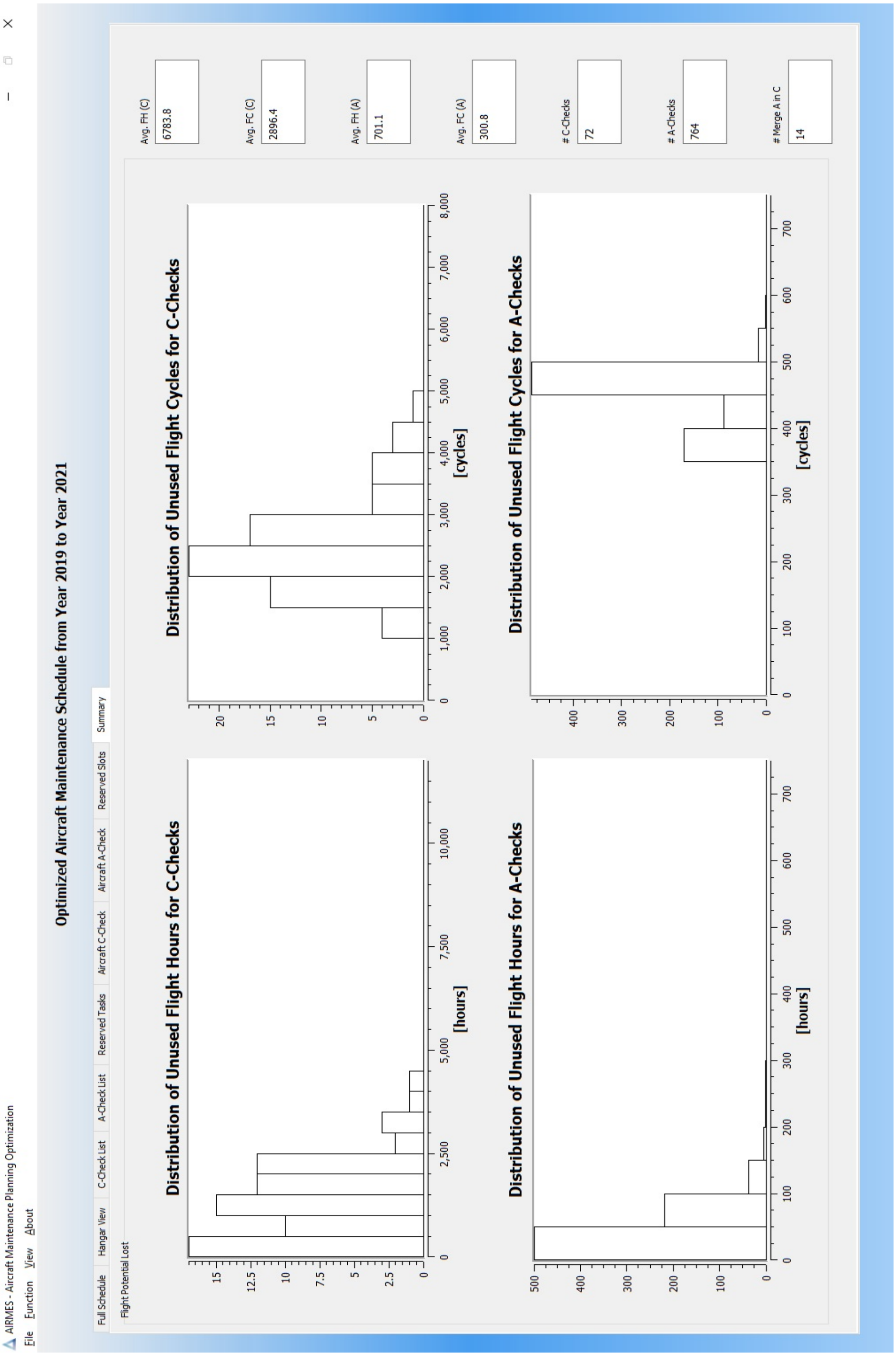


Figure 5.6: The KPIs of maintenance check schedule of the airline. We used the DSS to load the maintenance check schedule of the airline directly and visualized the results on the interface.

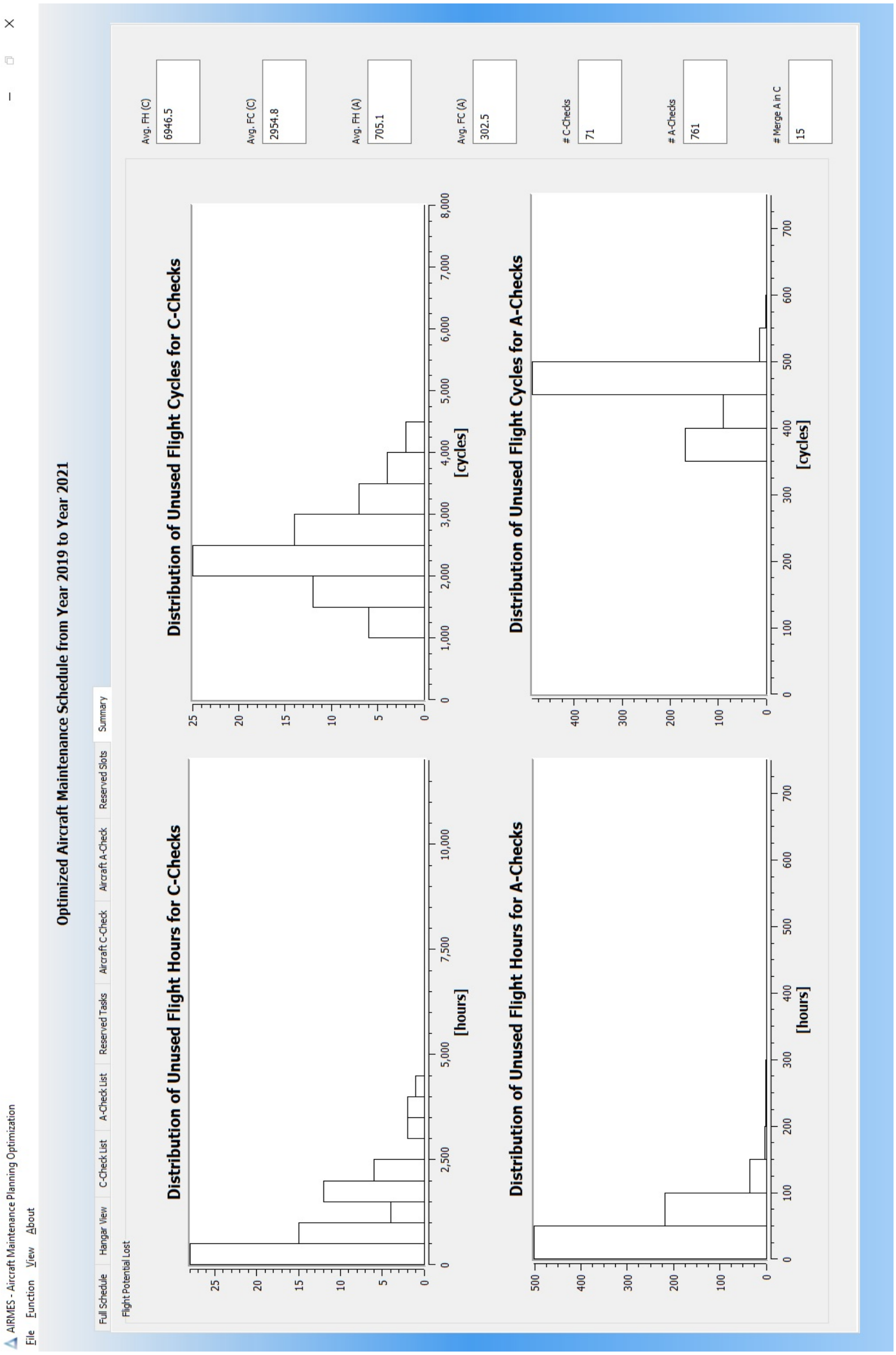


Figure 5.7: The KPIs of maintenance check schedule from the AMPO-1 of the DSS

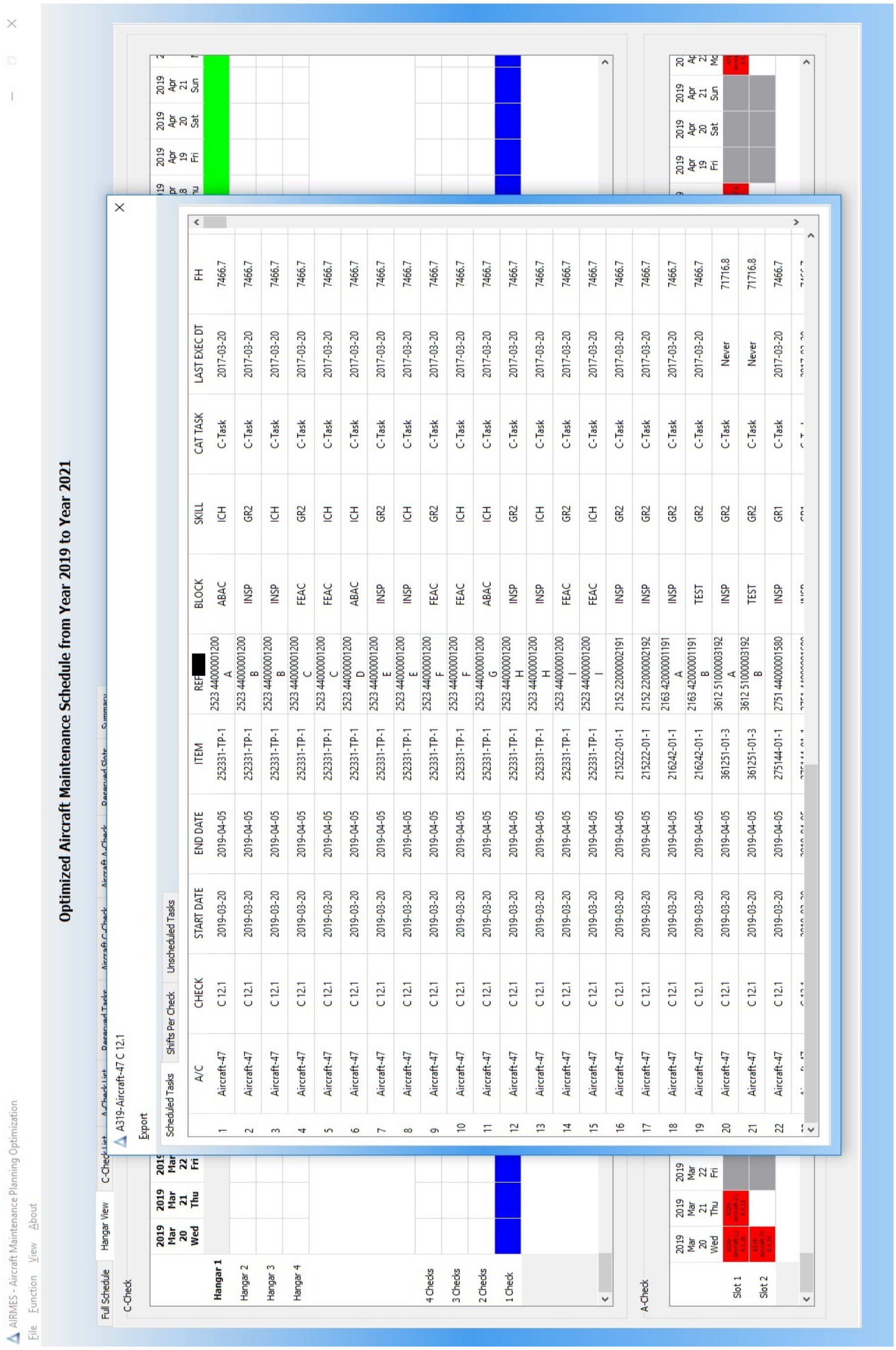


Figure 5.8: Results of optimal task allocation (AMPO-2).

Table 5.5: An example of work shifts planned by AMPO-3.

Aircraft	Date	Shift	Item	Block	Skill Type and Required Labor-Hours							
					S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8
AC-1	09-20-2018	Morning	Open panel '734'	—	0.6	0.2	0	0	0	0	0	0
AC-1	09-20-2018	Morning	Open panel '151KW'	—	0	0.2	0	0	0	0	0	0
AC-1	09-20-2018	Morning	Open panel '312AR'	—	0.1	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
AC-1	09-20-2018	Afternoon	251100-TP-1	INSP	0	0	0	0	2.0	0	0	0
AC-1	09-20-2018	Afternoon	575161-01-2	INSP	0.8	0	0	0.2	0.2	0.2	0.1	0
AC-1	09-20-2018	Afternoon	792000-C4-2	LUB	0.5	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
AC-1	09-20-2018	Night	575164-01-4	ABAC	0.5	0	0	0	0	0	0	0
AC-1	09-20-2018	Night	575164-01-4	INSP	0	0	0	0	0	0	0	0.5
AC-1	09-20-2018	Night	575164-01-4	FEAC	0.5	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

5

requires one labor-hour for a specific skill type, it can be one technician spending an hour, or two technicians spend half an hour, or even four technicians spend 15 minutes performing the task.

5.4.2. EVALUATION OF AIRCRAFT MAINTENANCE STRATEGIES

In the second test case, we used the DSS to evaluate different maintenance strategies before implementation. Each strategy is modeled as a test scenario, and all scenarios are compared to the baseline scenario (Airline Schedule). Three maintenance strategies (test scenarios) were proposed by the airline:

1. *Scenario 1*: increase the number of daily C-check slots from three to four but reducing the period of the year in which the C-checks can be performed from the current *October–May* to *November–March*;

2. *Scenario 2*: increase the fleet size from 51 to 66 aircraft without changing the maintenance periods or number of slots available;

3. *Scenario 3*: increase the fleet size from 51 to 66 aircraft but now increasing the A-check slots by one on Fridays.

Table 5.6 shows the KPIs from test scenarios. We also include the KPIs from the previous demonstration and use the airline schedule as the baseline scenarios. First of all, without considering other costs, we see the benefit per aircraft from implementing the DSS (DSS Schedule) compared with the *baseline* scenario (Airline Schedule) for a 3-year planning horizon:

$$\text{C-Check:} \qquad \underbrace{20.6}_{\text{gain}} + \underbrace{4.9}_{\text{saving}} - \underbrace{0}_{\text{cost}} = 30.9\text{K} \tag{5.3}$$

$$\text{A-Check:} \qquad \underbrace{5.7}_{\text{gain}} + \underbrace{0.7}_{\text{saving}} - \underbrace{0}_{\text{cost}} = 6.4\text{M} \tag{5.4}$$

The KPIs of *Scenario 1* indicate that shortening the C-check periods while increasing the C-check capacity, as considered by the airline, is not enough to cope with the C-check demand from the current fleet size and leads to a loss of $75.4K in total per aircraft for two check types. Although it gains $78.0K from more days of commercial operations compared with the *baseline* scenario (Airline Schedule) and saves $14.7K because of performing fewer C-checks, the airline needs to spend $66.0K per aircraft on creating extra C-check slots (non-existent daily slots are added to the schedule, representing moments that technicians have to work extra-time or that additional workforce has to be hired). The majority of loss comes from A-check due to grounding aircraft more often for A-checks (-$89.9K from commercial operations and -$11.5K for performing more A-check) and creating more extra A-check slots (-$0.7K). The reason is that the optimization algorithm of AMPO-1 tries to ground the aircraft for A-check more often to defer the need for a C-check. For example, consider an aircraft with a C-check interval of 7500 FH/730 DY and an average daily operation of 15 FH. If this aircraft has no A-check, it will be grounded and performed a C-check after 500 days since the FH usage parameter reaches 7500 FH before the DY usage parameter reaches 730 DY. If there is one A-check

Table 5.6: Summary of KPIs from the airline schedule (3rd column), the AMCS optimization for the first test case (4th column), and the different scenarios from the second test case (5th–7th column). The "Airline Schedule" serves as the baseline scenario. "Gain" represents the potential income generated per aircraft from having more days for commercial operations (due to more/few days for A- or C-checks) compared with the baseline scenario. "Saving" represents the reduction of maintenance costs per aircraft due to more/fewer checks. "Cost" represents the costs per aircraft for creating extra slots.

KPIs 20/03/2019–31/12/2021		**Airline Schedule**	**DSS Schedule**	**Scenario 1**	**Scenario 2**	**Scenario 3**
C-check	Average FH	6783.8	6946.5	6959.8	6543.8	6012.3
	Average FC	2896.4	2954.8	2955.4	2802.5	2570.2
	Total Checks	72	71	69	73	76
	Extra Slots	0	0	61	0	0
	Gain [$]	—	26.0K	78.0K	-20.1K	-80.4K
	Saving [$]	—	4.9K	14.7K	-3.8K	-15.2K
	Cost [$]	—	0	66.0K	0	0
A-check	Average FH	701.1	705.1	665.0	690.6	664.3
	Average FC	300.8	302.5	285.6	292.7	281.6
	Total Checks	764	761	811	929	967
	Extra Slots	3	3	4	75	9
	Gain [$]	—	5.7K	-89.9K	-243.8K	-299.9K
	Saving [$]	—	0.7K	-11.5K	-31.3K	-38.4K
	Cost [$]	—	0	0.7K	40.9K	3.4K
Total Benefit per Aircraft		—	37.3K	-75.4K	-339.8K	-437.3K

According to our airline partner:

1) One day of operation generates on average $97.5K of revenue.

2) The A-check of an A320 family aircraft lasts one working day and costs on average $12.5K.

3) The C-check of an A320 family aircraft lasts on average 13.6 working days (slots).

4) One fewer A-(C-)check means the entire fleet can have 1(13.6) more days for operations.

5) The C-check of an A320 family aircraft costs on average $250K ($18.4K per working day).

6) The cost of creating one extra A-/C-check slot is three times as one normal slot.

scheduled before the C-check (A-check lasts one day), this aircraft can have the C-check after 501 days. Similarly, if the aircraft is scheduled two A-checks, it can have the C-check after 502 days, and so forth. Based on the results of the *Scenario 1* evaluation, we suggested that the airline should use its current maintenance strategy rather than the new one (described in *Scenario 1*).

For *Scenario 2*, we observed that the current A-check capacity is not sufficient to handle the increased A320 fleet size according to the results shown in Table 5.6. Eventually, it leads to a huge loss, -\$339.8K, on average per aircraft. Apart from the loss from C-check due to fewer commercial operations (-\$20.1K) and performing more C-check (-\$3.8K), A-check contributes most to the loss, -\$243.8K from commercial operations and -\$31.3K from performing more A-checks. Besides, it needs to spend \$40.9K on average per aircraft on creating extra A-check slots.

To cope with the soaring A-check demand from increased fleet size, the airline proposes to add one A-check slot on Friday, as described in *Scenario 3*. According to the DSS results, creating one additional aircraft A-check slot on Friday (*Scenario 3*) significantly reduces the need for extra A-check capacity from 75 to 9 compared with *Scenario 2*, meaning that the cost of creating extra slots is reduced (from \$40.9K to \$3.4K). However, it also increases the number of checks for both check types, resulting in a huge revenue loss from commercial operations. The total loss increases by $437.3 - 339.8 = 97.5$K on average per aircraft compared with *Scenario 2*. We found out that the optimization algorithm schedules C-check more frequently to provide more opportunities to merge the A-checks in C-checks (since the airline primarily wanted to avoid creating extra slots). Based on the results of *Scenario 2* and *Scenario 3*, we suggested that adding one A-check slot per week is not sufficient for the increase of fleet size, and the airline should consider adding more A-check slots to cope with the increased maintenance check demand.

5.5. CONCLUSION

This paper presents a decision support system (DSS) that addresses aircraft maintenance planning optimization in an integrated fashion, automating repetitive tasks while enabling fast, efficient, human-in-the-loop decision making for optimized planning purposes. First of all, the DSS is capable of optimizing the aircraft maintenance check schedule. Secondly, based on the optimal maintenance check schedule, the DSS allocates maintenance tasks to each maintenance check considering the overlapping situation (having multiple checks on going in the same period). Thirdly, the DSS plans the work shift respecting the task sequence in practice. It can potentially help airlines improve their maintenance planning efficiency, reduce the related maintenance operation costs, and even assess their maintenance strategies. Therefore, the DSS makes significant contributions relevant to both scientific research and industry application.

The DSS bridges the gap between long-term AMCS and short-term shift planning. It integrates aircraft maintenance check scheduling, maintenance task allocation and, work shift planning in the same platform. A demonstration exercise with a European airline shows that the DSS can generate a comprehensive optimal maintenance plan for a planning horizon of three years within half an hour. It means airlines can use the DSS to reduce the time needed for aircraft maintenance planning from several days to about 30 minutes. More importantly, considering the uncertainty that might impact aircraft utilization or maintenance activities, we make it possible for the maintenance planners to run the DSS in a short time to update the current plan. Whenever there are changes in the aircraft maintenance tasks or maintenance activities, maintenance planners can quickly make new decisions using the DSS and re-organize the tools, workforce or promptly prepare the aircraft spare parts.

Besides, the demonstration exercise results show that the DSS reduces the number of A-/C-checks by three/one while increasing the expected average FH of A-/C-check by 2.4%/0.6% for a planning horizon of 3 years compared with the maintenance check schedule made using the planning approach of the airline. The reduced A- and C-checks could lead to a significant saving in maintenance costs. The improved aircraft utilization also indicates that there will be more aircraft available for day-to-day commercial operations to generate additional revenue for airlines. After the demonstration exercise, the DSS was tested and classified by the Clean Sky Joint Undertaking partners to be at a Technology Readiness Level Six (TRL 6). Nevertheless, the tool still has some limitations that have to be addressed in the future if a higher TRL is aimed:

- The primary goal of AIRMES was on the development of the optimization algorithms, so future efforts should focus on improving the GUI.
- Define requirements and specifications that will facilitate direct integration of the DSS with other information systems used by airlines, including the development of the Application Programming Interface (API) and, potentially, a Software Development Kit (SDK).
- Include the number of aircraft spare parts in the constraints in the task allocation (AMPO-2) model.

Another interesting direction is to incorporate condition-based maintenance (CBM) by taking health prognostics and diagnostics into consideration when developing maintenance plans. Although including CBM in the DSS will increase model complexity and computation time, it will prepare the tool to cope with a current trend in the aircraft maintenance research and operational communities.

Finally, it is worth mentioning that although the DSS is tailored to aircraft maintenance planning optimization, it can also be adjusted to address similar problems, such as train or bus maintenance planning for the coming days or weeks, or to match the maintenance demand with operation timetables. For example, the main screen of the DSS can be changed to display daily operation hours and maintenance duration. The algorithm described in [5] can be adapted for similar maintenance scheduling or even more general scheduling problems (e.g., vehicle routing or production planning) since the idea of the algorithm is to estimate the consequence of each possible (maintenance) action before making a decision. For such applications, the DSS framework can be maintained.

REFERENCES

[1] S. P. Ackert, *Basics of Aircraft Maintenance Programs for Financiers,* (2010), (Accessed on September 28, 2017).

[2] IATA's Maintenance Cost Task Force, *Airline Maintenance Cost Executive Commentary Edition 2019,* (2019), (Accessed on September 11, 2020).

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

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

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

[6] T. A. Feo and J. F. Bard, *Flight Scheduling and Maintenance Base Planning,* Management Science **35**, 1415 (1989).

[7] J. L. Higle and A. E. C. Johnson, *Flight Schedule Planning with Maintenance Considerations,* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.542.3117&rep=rep1&type=pdf (2005), (Accessed on March 5, 2021).

[8] L. Clarke, E. Johnson, G. Nemhauser, and Z. Zhu, *The aircraft rotation problem,* Annals of Operations Research **69**, 33 (1997).

[9] M. Başdere and U. Bilge, *Operational aircraft maintenance routing problem with remaining time consideration,* European Journal of Operational Research **235**, 315 (2014).

[10] Z. Liang, W. A. Chaovalitwongse, H. C. Huang, and E. L. Johnson, *On a new rotation-tour network model for aircraft maintenance routing problem,* Transportation Science **45**, 109 (2011).

[11] C. Barnhart, N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, and R. G. Shenoi, *Flight String Models for Aircraft Fleeting and Routing,* Transportation Science **32**, 208 (1998).

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

[13] Z. Liang and W. A. Chaovalitwongse, *A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem,* Transportation Science **47**, 493 (2013).

[14] C. Sriram and A. Haghani, *An Optimization Model for Aircraft Maintenance Scheduling and Re-Assignment,* Transportation Research Part A **37**, 29 (2003).

[15] S. Gabteni and M. Grönkvist, *A Hybrid Column Generation and Constraint Programming Optimizer for the Tail Assignment Problem,* in *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming* (2006).

[16] N. Safaei and A. K. Jardine, *Aircraft routing with generalized maintenance constraints*, Omega **80**, 111 (2018).

[17] A. M. Cohn and C. Barnhart, *Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions*, Operations Research **51**, 343 (2003).

[18] N. Papadakos, *Integrated airline scheduling*, Computers & Operations Research **36**, 176 (2009).

[19] J. Díaz-Ramírez and J. I. H. F. Trigos, *Aircraft maintenance, routing, and crew scheduling planning for airlines with a single fleet and a single maintenance and crew base*, Computers & Industrial Engineering **75**, 68 (2014).

[20] M. C. Dijkstra, L. G. Kroon, M. Salomon, J. A. E. E. V. Nunen, and L. N. V. Wassenhove, *Planning the Size and Organization of KLM's Aircraft Maintenance Personnel*, Interfaces **24**, 47 (1994).

5

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

[22] T.-H. Yang, S. Yan, and H.-H. Chen, *An airline maintenance manpower planning model with flexible strategies*, Journal of Air Transport Management **9**, 233 (2003).

[23] D. E. Ighravwe and S. A. Oke, *A non-zero integer non-linear programming model for maintenanceworkforce sizing*, International Journal of Production Economics **150**, 204 (2014).

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

[25] C. I. Permatasari, Yuniaristanto, W. Sutopo, and M. Hisjam, *Aircraft maintenance manpower shift planning with multiple aircraft maintenance licenced*, in *IOP Conference Series:: Materials Science and Engineering 495 012023* (2019).

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

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

[28] J. Beliën, E. Demeulemeester, Philippe, D. Bruecker, J. V. den Bergh, and B. Cardoen, *Integrated staffing and scheduling for an aircraft line maintenance problem*, Computers & Operations Research **40**, 1023 (2013).

[29] N. Safaei, D. Banjevic, and A. K. S. Jardine, *Workforce-constrained maintenance scheduling for military aircraft fleet: a case study*, Annals of Operations Research **186**, 295 (2011).

[30] M. Witteman, Q. Deng, and B. F. Santos, *A Bin Packing Approach to Solve the Aircraft Maintenance Task Allocation Problem,* European Journal of Operational Research (2021), https://doi.org/10.1016/j.ejor.2021.01.027, accepted.

[31] AIRMES, *AIRMES Demonstration Video,* (2019), (Accessed on July 5, 2020).

[32] Clean Sky, *Digitalisation of aircraft maintenance with Clean Sky 2's AIRMES project,* (2019), (Accessed on July 5, 2020).

[33] B. Kaupp, C. Mostert, N. Bontikous, and H. Meyer, *E2E Results and Final Project Report*, Tech. Rep. (Clean Sky 2, 2020) Private Document.

6

CONCLUSION

This chapter concludes the dissertation with observations and findings obtained during the study of aircraft maintenance planning optimization. First of all, it reviews the main objectives of this research work. Next, it describes the challenges met in aircraft maintenance planning and how they are addressed eventually, to highlight the scientific novelties and practical contributions. Finally, this chapter discusses the limitations of the aircraft maintenance planning models and gives recommendations for future work.

6.1. REVIEW OF RESEARCH OBJECTIVE

This dissertation is dedicated to aircraft maintenance planning optimization. As stated in Chapter 1, the main research objective is:

> *To develop a comprehensive maintenance planning optimization framework, including aircraft maintenance check scheduling and the associated maintenance task allocation, that automates and optimizes the aircraft maintenance planning process without compromising the long-term efficiency.*

Since this dissertation divides the main objective two sub-objectives, in this section, each sub-objective and the conclusions reached from the research are reviewed.

O-1 Optimize the aircraft maintenance check schedule and task execution plan.

- Optimize the long-term deterministic aircraft maintenance check schedule,
- Optimize the task execution plan for each maintenance check, and
- Optimize the aircraft maintenance check decision considering uncertainties.

Chapters 2–4 of this dissertation are dedicated to describing how **O-1** is achieved from three different aspects, given a specific aircraft fleet and time horizon, and each chapter corresponds to one particular topic. Chapter 2 focuses on the long-term deterministic aircraft maintenance check scheduling (AMCS) optimization. It presents a formulation that includes all letter check types in the same model and considers detailed real-life operational constraints. Moreover, Chapter 2 presents a dynamic programming (DP) based methodology for the long-term deterministic AMCS. A case study with a European airline shows that, compared with current practice, the DP-based methodology reduces the total number of letter checks, potentially resulting in a maintenance cost saving of about \$1.1M–\$3.4M for a fleet of about 40 aircraft and a 4-year horizon.

Chapter 3 addresses the task allocation problem (TAP) of each maintenance check. It formulates the TAP as a time-constrained variable-size bin packing problem (TC-VS-BPP), given an optimal aircraft maintenance check schedule. Each bin is a time period when several aircraft have the same type of letter checks ongoing in parallel and share the maintenance resources. The bin size is the associated available workforce within this period. A constructive heuristic based on the worst-fit decreasing (WFD) algorithm is proposed, aiming at minimizing the total labor costs. According to a real-life case study on 45 aircraft, the heuristic is more than 30% faster than an exact method (addressed by a commercial optimization solver), while the solution gap is smaller than 0.1%.

Chapter 4 presents a lookahead approximate dynamic programming (ADP) methodology for the stochastic AMCS optimization, considering the uncertainty of aircraft daily utilization and maintenance elapsed time. The lookahead ADP methodology consists of a DP framework and a hybrid lookahead policy with deterministic and stochastic forecasts. 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, given current decisions, using Monte Carlo simulations. Case studies of an AIRBUS A320 fleet show that the lookahead ADP methodology determines the daily optimal maintenance check decision only in a few seconds. The lookahead ADP methodology can be used by the maintenance operators to quickly update aircraft maintenance check decisions whenever changes occur in maintenance tasks or activities.

Since the dissertation has provided solutions for all three aspects of **O-1**, meaning that it addresses the long-term AMCS optimization and the associated optimal task allocation, and the stochastic AMCS optimization, following the top-down approach commonly used in the aviation industry. Therefore, the Sub-Objective **O-1**, optimizing the aircraft maintenance check schedule and task execution plan, is successfully achieved.

O-2 Automate the aircraft maintenance planning process

Chapter 5 describes a decision support system (DSS) developed during this research work (AIRMES project). On the one hand, the DSS serves as a model framework to facilitate aircraft maintenance planning. The DSS users (maintenance operators of airlines) no longer need to use different tools for AMCS and task allocation separately. On the other hand, it automates the planning process by integrating the long-term deterministic AMCS optimization and the associated optimal task allocation, and the stochastic AMCS optimization in the same framework. Besides, the shift planning function is also added in the last phase of DSS development. After the DSS users load the input data via the user interface, the DSS automatically optimizes the aircraft maintenance check schedule and the associated task execution activities and plan the work shifts. It significantly improves the aircraft maintenance planning efficiency since maintenance operators of airlines do not need to move the maintenance checks/tasks manually or worry about the feasibility of a maintenance check schedule. After a demonstration exercise, the DSS was tested and classified by the Clean Sky Joint Undertaking partners to be at a Technology Readiness Level Six (TRL 6). Hence, this dissertation also reaches the Sub-Objective **O-2**, automating the aircraft maintenance planning process.

6

6.2. Research Novelty and Practical Contribution

This dissertation presents not only methodologies that can potentially innovate aircraft maintenance planning (AMP) but also a decision support system (DSS) that can be used to automate the AMP process. There were many difficulties during the research of the aircraft maintenance planning (AMP) optimization, and they were addressed as follows:

- *There is a lack of data about aircraft maintenance check scheduling (AMCS) and seasonal daily aircraft utilization, and these data are crucial to formulating the AMCS problem and developing optimization algorithms*:

  The first scientific contribution of this dissertation is to collect and anonymize the AMCS related data, including fleet status, aircraft daily utilization, maintenance check capacity and peak seasons, and share those data with the public. This dissertation provides researchers access to AMCS-related data for further study of the AMCS or similar problems.

- *During the algorithm development for the deterministic AMCS optimization, the author found out that the action space (the combinations of selecting multiple air-*

craft for maintenance checks) and solution space are extremely large (the final optimal fleet status is unknown), making it difficult to keep track of the AMCS decisions. Besides, there are no available cost data to model the impacts of an AMCS decision.

For the first time, this dissertation proposes to assign priorities to aircraft for maintenance checks according to aircraft utilization following the rule of “earliest deadline first”, which significantly reduces the size of the action space. It is also the first to adopt forward induction under dynamic programming (DP) framework to optimize the AMCS decisions for multiple check types. To keep track of the AMCS decisions, it adapts the classic discretization and state aggregation of the DP approach to make the AMCS problem tractable. Instead of using the cost data to model the impacts of an AMCS decision, this dissertation uses a thrifty algorithm to check whether the future maintenance capacities can cope with the demands so that whenever there are sufficient capacities for all check types, the thrifty algorithm will suggest deferring a maintenance check action. All these innovations make the proposed DP-based methodology described in Chapter 2 applicable to real-life problems. A case study of a European airline shows that the DP-based methodology optimized the maintenance check schedule for 45 aircraft and a 4-year planning horizon within 15 minutes.

- *The long-term aircraft maintenance task allocation problem (TAP) has never been addressed before due to the lack of an optimal aircraft maintenance check schedule.*

6

This dissertation proposes a constructive heuristic based on the worst-fit decreasing (WFD) algorithm that optimizes the maintenance task execution within each maintenance check quickly. Given a 4-year optimized aircraft maintenance check schedule for 45 aircraft, the constructive heuristic described in Chapter 3 determines the start date of each aircraft maintenance task in 20 minutes while the optimality gap from a solution obtained by a commercial solver is within 5%. Whenever there are changes in aircraft maintenance tasks or maintenance activities, the proposed constructive heuristic can promptly update the maintenance task execution and the corresponding workforce.

- *There are uncertainties in the aircraft maintenance check elapsed and in the aircraft daily utilization. The optimal solution from the deterministic AMCS model may not be robust and require to be updated frequently.*

This dissertation presents a lookahead approximate dynamic programming (ADP) for the stochastic AMCS optimization. The lookahead ADP still adopts the DP framework. It uses a hybrid lookahead scheduling policy first to make the optimal decision for heavy aircraft maintenance (e.g., C-/D-check) based on deterministic forecasts, i.e., examining whether the capacity in a predefined future time-window is sufficient for the maintenance demands using the deterministic maintenance check elapsed time and aircraft daily utilization. After that, it fixes the decisions for heavy maintenance and determines the light maintenance (e.g., A-/B-check) according to stochastic forecasts, i.e., checking whether the light maintenance demands can fit in the existing available maintenance slots using Monte Carlo simulations. A case study of 45 aircraft shows that, compared with the current practice, the lookahead ADP 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.

- *There is a lack of efficient tools for aircraft maintenance planning in the aviation industry. Airlines have to generate the aircraft maintenance check schedule and plan the maintenance task execution separately.*

  This dissertation also contributes to the state-of-the-art development of aircraft maintenance planning software. It presents a decision support system (DSS) that integrates the AMCS, maintenance task allocation, and shift planning in the same optimization framework. This novel DSS automates the repetitive maintenance planning process while enabling fast, efficient, human-in-the-loop decision making for optimized planning purposes. It can not only optimize the aircraft maintenance check schedule considering the dependence of different check types and the associated task execution but also plan the work shift respecting the task sequence in practice. A demonstration exercise with an airline partner shows that the DSS is capable of generating a comprehensive optimal maintenance plan for a planning horizon of three years within half an hour, successfully reducing the time needed for aircraft maintenance planning from several days to about 30 minutes. The DSS was tested and classified by the Clean Sky Joint Undertake partners to be at a Technology Readiness Level Six (TRL-6).

6

6.3. Research Limitation and Recommendations

This dissertation has provided significant contributions to both scientific research and industry application. Even so, there are still some impacts or challenges that have not been considered in the research work:

- Aircraft line maintenance planning

Aircraft line maintenance refers to the activities carried out while the aircraft remains in the operating environment and is airworthy to fly. Aircraft line maintenance is usually performed at the gate. Line maintenance tasks include replacement of any component designated as a line replaceable unit, routine in-service inspections, and day-to-day check actions, and so on. Although the AMCS models and corresponding methodologies proposed in this dissertation potentially reduce the number of maintenance checks and increase average aircraft utilization, this, on the other hand, may transfer more workloads to line maintenance and increase line maintenance costs as a result. Therefore, including aircraft line maintenance in the AMP optimization can further reduce the total maintenance operation costs for airlines. Following this direction, one can continue AMP to line maintenance planning optimization, i.e., determine the tasks to be executed in line maintenance or further move some tasks within aircraft letter check (A-/B-/C-/D-check) to line maintenance. In this way, it reduces not only the total maintenance operation costs but also the risk of spending more time on a maintenance check than planned.

- Optimizing the composition and performance of teams of aircraft maintenance technicians

In the shift planning model, one of the goals is to assign the right number of technicians to execute a maintenance task. However, due to the lack of data, the airline partner suggests a maximum of four technicians for the execution of one maintenance task. In practice, the number of technicians in a specific area of an aircraft is limited. It is not realistic to allow as many technicians as possible to work on a task at a time. In this case, it is also necessary to collect data regarding the number of technicians performing the same job. Including this information in the shift planning mode, one can optimize the number of technicians per shift and even further plan the compositions of teams per activity scheduled.

- Considering uncertainties in the task allocation model

This dissertation only considers the uncertainties from aircraft daily utilization and maintenance check elapsed time in the stochastic AMCS model, at the maintenance check level. The task allocation model does not include uncertainties at the maintenance task level. Incorporating task delays or possible system or component failures in the task allocation model is also an exciting research direction. One potential solution is to introduce condition-based maintenance (CBM). The CBM uses health prognostics and diagnostics to define the tasks within each maintenance check. With the CBM approach, the task allocation model can plan the maintenance tasks for each maintenance check according to real-time monitoring rather than fixed intervals, further extending the lives of aircraft components and systems.

- Improving the applicability of the DSS

Regarding practical implementation, although the DSS is demonstrated to be efficient in AMP optimization, there are still some possible improvements to make the DSS applicable. The future works for the DSS improvements are in two aspects. On the one hand, the graphical user interface (GUI) still needs some effort to be more user-friendly and resilient. On the other hand, direct integration of the DSS with other information systems used by airlines requires developing an Application Programming Interface (API) and, potentially, a Software Development Kit (SDK) to facilitate this integration.

- Combining preventive maintenance and condition-based maintenance

The AMCS optimization and the associated optimal aircraft maintenance task allocation belong to preventive maintenance (PM) scheduling. In PM scheduling, one has to first estimate when a maintenance check or task is due according to its inspection interval and daily utilization, then set a date, time, and place, and assign technicians to perform the work. In condition-based maintenance (CBM), the maintenance tasks are planned according to health prognostics and diagnostics. The design of the stochastic AMCS model allows the integration of CBM and under AMCS. In this case, one can continue to develop health monitoring algorithms to further divide the letter checks into more frequent and smaller task blocks that give more flexibility and allow quicker reactions to new information.

Curriculum Vitæ

Qichen Deng

Qichen Deng (邓启辰 in Chinese) was born on May 11th, 1987, in Guangdong, China. He received his Bachelor's degree in Mathematics from South China University of Technology (China) in 2009 and Master's degree in Optimization and System Theory from KTH Royal Institute of Technology (Sweden) in 2013. Afterward, he continued to work as a research engineer in the Division of Traffic and Logistic, KTH Royal Institute of Technology, while pursuing his postgraduate study in both KTH Royal Institute of Technology and the National Intelligent Transportation System (ITS) Research School of Sweden.

From 2013 to 2016, Mr. Deng participated in the project iQFleet — Intelligent Real-Time Fleet Control and Management, collaborating with one of the leading truck manufacturers, Scania. The iQFleet project was funded by Vinnova (Sweden's Innovation Agency). Mr. Deng was responsible for simulations of the heavy-duty vehicle (HDV) platooning strategies. He developed a simulation platform to implement platooning control systems and study the impact of HDV platooning on traffic flow.

After he obtained the Licentiate Degree from both KTH Royal Institute of Technology and the National ITS Research School of Sweden, Mr. Deng became a PhD student at the Section of Air Transport and Operations, Delft University of Technology (TU Delft), funded by the European Union's Horizon 2020 Research and Innovation Programme. As part of his Ph.D., he participated in and contributed to the results of the EU Project AIRMES. His research focused on aircraft maintenance planning optimization for commercial fleet, which aims at helping airlines to improve aircraft maintenance efficiency and reduce maintenance operation costs. He provided solutions for the long-term aircraft maintenance check scheduling optimization and optimal maintenance task allocation. His scientific contributions have been presented in local and international conferences, leading to several publications in a few top Operations Research related journals. Alongside the research, he has undertaken educational tasks to supervise Master students. Mr. Deng was the daily supervisor of three master theses ("Maintenance Planning and Scheduling for Aircraft Line Maintenance", "A Practical Maintenance Task Packaging Model Applicable to Aircraft Maintenance", and "Long-Term C-Check Scheduling for A Fleet of Heterogeneous Aircraft under Uncertainty"). Apart from doing research, he is a martial artist, and he practices piano in his leisure time.

List of Publications

Publications Related to The Dissertation

4. **Q. Deng** and B. F. Santos, *Lookahead Approximate Dynamic Programming for Stochastic Aircraft Maintenance Check Scheduling Optimization*, submitted to European Journal of Operational Research in 2021 [unpublished-under review].

3. **Q. Deng** and B. F. Santos and W. J. C. Verhagen, *A Novel Decision Support System for Optimizing Aircraft Maintenance Check Schedule and Task Allocation*, Decision Support Systems, 2021 (DOI: https://doi.org/10.1016/j.dss.2021.113545).

2. M. Witteman, **Q. Deng**, and B. F. Santos, *A Bin Packing Approach to Solve the Aircraft Maintenance Task Allocation Problem*, European Journal of Operational Research, 2021 (DOI: https://doi.org/10.1016/j.ejor.2021.01.027).

1. **Q. Deng**, B. F. Santos, and R. Curran, *A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization*, European Journal of Operational Research, **Volume 281**, **Issue 2**, pp. 256-273 (2020).

Other Publications in Peer-Reviewed Journals

2. T. M. J. van der Weide, **Q. Deng**, and B. F. Santos, *Robust Long-Term Aircraft Heavy Maintenance Check Scheduling Optimization under Uncertainty*, submitted to Computers & Operations Research in 2021 [unpublished-under review].

1. **Q. Deng**, *A General Simulation Framework for Modeling and Analysis of Heavy-Duty Vehicle Platooning*, IEEE Transactions on Intelligent Transportation Systems, **Volume 17**, **Issue 11**, pp. 3252-3262 (2016).