ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διατριβή

ΒΕΛΤΙΣΤΗ ΚΑΤΑΝΟΜΗ ΩΡΩΝ ΠΤΗΣΗΣ ΚΑΙ ΣΥΝΤΗΡΗΣΗΣ ΣΕ ΕΠΙΧΕΙΡΗΣΙΑΚΑ ΑΕΡΟΣΚΑΦΗ. ΜΟΡΦΟΠΟΙΗΣΕΙΣ, ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ

υπό

ΑΝΔΡΕΑ ΓΑΒΡΑΝΗ

Μηχανικού Τηλεπικοινωνιών και Ηλεκτρονικών Σχολής Ικάρων, 2003

ΜΔΕ Μηχανολόγων Μηχανικών Πανεπιστημίου Θεσσαλίας, 2008

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Διδακτορικού Διπλώματος

2017

Η έγκριση της διδακτορικής διατριβής από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Επταμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων)	Δρ. Γεώργιος Κοζανίδης Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Γεώργιος Λυμπερόπουλος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Δρ. Αθανάσιος Ζηλιασκόπουλος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τέταρτος Εξεταστής	Δρ. Γιώργος Σαχαρίδης Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Πέμπτος Εξεταστής	Δρ. Μιχαήλ Δούμπος Αναπληρωτής Καθηγητής, Σχολή Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης
Έκτος Εξεταστής	Δρ. Αθανάσιος Μυγδαλάς Καθηγητής, Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Sweden
Έβδομος Εξεταστής	Δρ. Χρήστος Ταραντίλης Καθηγητής, Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας, Οικονομικό Πανεπιστήμιο Αθηνών

Ευχαριστίες

Πρώτα απ’ όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διδακτορικής μου διατριβής, Επίκουρο Καθηγητή Γεώργιο Κοζανίδη, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια εκπόνησής της. Επίσης, είμαι ευγνώμων στους καθηγητές κκ Γεώργιο Λυμπερόπουλο και Αθανάσιο Ζηλιασκόπουλο για τις πολύτιμες γνώσεις που μού προσέφεραν κατά τη διάρκεια των σπουδών μου, καθώς και στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διδακτορικής μου διατριβής, Καθηγητές κκ Γεώργιο Σαχαρίδη, Μιχάλη Δούμπο, Αθανάσιο Μυγδαλά και Χρήστο Ταραντίλη για την προσεκτική ανάγνωση της εργασίας μου. Οφείλω ευχαριστίες σε όλους τους κατά καιρούς καθηγητές μου, οι οποίοι συνετέλεσαν καθοριστικά στην απόκτηση του θεωρητικού μου υποβάθρου, το οποίο αποδείχτηκε πολύτιμο για την ολοκλήρωση της παρούσας, ιδιαίτερα απαιτητικής, διατριβής. Είμαι πραγματικά ευγνώμων στους γονείς μου, Θανάση και Μένη, και στην αδερφή μου Ασπελίνα για την ολόψυχη αγάπη και την αέναη υποστήριξή τους. Πάνω απ’ όλα, ευχαριστώ τη σύζυγό μου Μαρία για την κατανόησή της, ιδιαίτερα κατά την τελευταία περίοδο της προσπάθειάς μου, όταν και έφερε στον κόσμο τον πρωτότοκο γιο μας, καθώς και τη μητέρα της Κωνσταντίνα για την πολύτιμη βοήθειά της. Στο γιο μου Αθανάσιο υπόσχομαι να είμαι ο πρώτος που θα τού διδάξει την αξία της εκπαίδευσης και της κριτικής σκέψης με την προσδοκία ότι αυτή η διατριβή θα αποτελέσει για εκείνον πηγή έμπνευσης και λαμπρό παράδειγμα στην μετέπειτα ζωή του, έτσι ώστε πάντοτε να προσπαθεί και να εργάζεται σκληρά για κάθε τι που θα θελήσει να αποκτήσει.

Αφιερώνω αυτήν τη διατριβή στους γονείς μου, στη σύζυγό μου και στο γιο μου.

Ανδρέας Γαβράνης

ΒΕΛΤΙΣΤΗ ΚΑΤΑΝΟΜΗ ΩΡΩΝ ΠΤΗΣΗΣ ΚΑΙ ΣΥΝΤΗΡΗΣΗΣ ΣΕ ΕΠΙΧΕΙΡΗΣΙΑΚΑ ΑΕΡΟΣΚΑΦΗ. ΜΟΡΦΟΠΟΙΗΣΕΙΣ, ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ

ΑΝΔΡΕΑΣ ΓΑΒΡΑΝΗΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2017

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Κοζανίδης, Επίκουρος Καθηγητής Βελτιστοποίησης Συστημάτων Παραγωγής / Υπηρεσιών

Περίληψη

Η παρούσα διατριβή πραγματεύεται την ανάπτυξη μοντέλων μαθηματικού προγραμματισμού και αλγορίθμων βελτιστοποίησης για το πρόβλημα του σχεδιασμού του πτητικού έργου και των εργασιών περιοδικής συντήρησης επιχειρησιακών αεροσκαφών (στρατιωτικών ή πυροσβεστικών αεροσκαφών, ελικοπτέρων έρευνας και διάσωσης, κτλ.). Το πρόβλημα ανακύπτει στην καθημερινή λειτουργία μιας πτέρυγας επιχειρησιακών αεροσκαφών της Πολεμικής Αεροπορίας. Τα μαθηματικά μοντέλα που αναπτύσσονται προέρχονται από την ευρύτερη περιοχή της επιχειρησιακής έρευνας (γραμμική, μη γραμμική, ακέραια, και πολυκριτήρια βελτιστοποίηση), ενώ οι αλγόριθμοι επίλυσης είναι τόσο ευρετικοί όσο και αναλυτικοί, έτσι ώστε να παρέχεται ένα ικανοποιητικό αντιστάθμισμα ανάμεσα στην ποιότητα των παραγόμενων λύσεων και στους υπολογιστικούς πόρους που απαιτούνται για την εύρεση των λύσεων αυτών.

Αρχικά, έγινε η μορφοποίηση του προβλήματος με τη χρήση τεχνικών μοντελοποίησης μαθηματικού προγραμματισμού. Για το σκοπό αυτό, αναπτύχθηκαν διάφορα εναλλακτικά μοντέλα μεικτού ακέραιου προγραμματισμού, τα οποία διαφοροποιούνται ως προς την αντικειμενική συνάρτηση (στόχο) που χρησιμοποιούν ως μέτρο απόδοσης, αλλά και ως προς τους περιορισμούς που υιοθετούν για τον καθορισμό των εφικτών λύσεων του προβλήματος. Η μελέτη των μοντέλων αυτών κατέδειξε ότι, αν και η εφαρμογή τους οδηγεί στην εύρεση ολικά βέλτιστων λύσεων, οι υπολογιστικές τους απαιτήσεις είναι συχνά απαγορευτικές για προβλήματα ρεαλιστικού μεγέθους. Η παρατήρηση αυτή ώθησε τη σχετική έρευνα στην ανάπτυξη εξειδικευμένων αλγορίθμων για την επίλυση του προβλήματος. Προς την κατεύθυνση αυτή, αναπτύχθηκαν αρχικά ευρετικοί αλγόριθμοι επίλυσης, οι οποίοι έχουν την ικανότητα να προσεγγίζουν την ολικά βέλτιστη λύση με χαμηλές υπολογιστικές απαιτήσεις, χωρίς όμως να παρέχουν εγγυήσεις για την εύρεσή της. Στη συνέχεια, η σχετική έρευνα στράφηκε στην ανάπτυξη αναλυτικών αλγορίθμων επίλυσης, έτσι ώστε να καταστεί εφικτή η εύρεση

της ολικά βέλτιστης λύσης του προβλήματος. Για το σκοπό αυτό, αναπτύχθηκαν 4 τέτοιοι αλγόριθμοι. Ο πρώτος μπορεί να εφαρμοστεί όταν ο χρονικός ορίζοντας σχεδιασμού αποτελείται από μία χρονική περίοδο, ενώ ο δεύτερος όταν αποτελείται από πολλές. Οι άλλοι δύο αλγόριθμοι ενσωματώνουν μία επιπλέον αντικειμενική συνάρτηση, έτσι ώστε εκτός από τη μεγιστοποίηση της διαθεσιμότητας των αεροσκαφών να επιτυγχάνεται και η ελαχιστοποίηση της μεταβλητότητάς της. Αυτή είναι μία σημαντική λειτουργική απαίτηση του εξεταζόμενου προβλήματος, καθώς οδηγεί σε επιχειρησιακή ετοιμότητα η οποία δεν μεταβάλλεται σημαντικά από περίοδο σε περίοδο.

Για την υλοποίηση των προτεινόμενων μοντέλων μαθηματικού προγραμματισμού και αλγορίθμων επίλυσης, καθώς και για την εκτέλεση των σχετικών υπολογιστικών πειραμάτων, χρησιμοποιήθηκαν τα εμπορικά λογισμικά βελτιστοποίησης IBM ILOG CPLEX και LINGO, καθώς και η γλώσσα προγραμματισμού C/C++. Η ανάλυση των αποτελεσμάτων που προέκυψαν από την εκτέλεση των υπολογιστικών πειραμάτων κατέδειξε ότι οι επιδόσεις των εξειδικευμένων αλγορίθμων που αναπτύχθηκαν είναι σαφώς ανώτερες από τις επιδόσεις εμπορικών λογισμικών βελτιστοποίησης που μπορούν να χρησιμοποιηθούν εναλλακτικά για την επίλυση του προβλήματος. Ως εκ τούτου, οι εν λόγω αλγόριθμοι καθιστούν εφικτή την αποτελεσματική αντιμετώπιση προβλημάτων ρεαλιστικού μεγέθους.

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

Dissertation

OPTIMAL ALLOCATION OF FLIGHT AND MAINTENANCE HOURS TO MISSION AIRCRAFT. FORMULATIONS, SOLUTION ALGORITHMS AND COMPUTER IMPLEMENTATIONS

by

ANDREAS GAVRANIS

B.Sc. in Electronics and Telecommunications Engineering,

Hellenic Air Force Academy, 2003

M.Sc. in Mechanical Engineering, University of Thessaly, 2008

Submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

2017

The approval of this PhD Dissertation by the Department of Mechanical Engineering of the School of Engineering of the University of Thessaly does not imply acceptance of the writer's opinions (Law 5343/32 article 202 par.2).

Approved by:

First Examiner (Supervisor)	Dr. George Kozanidis Assistant Proffesor, Department of Mechanical Engineering, University of Thessaly
Second Examiner	Dr. George Liberopoulos Professor, Department of Mechanical Engineering, University of Thessaly
Third Examiner	Dr. Athanasios Ziliaskopoulos Professor, Department of Mechanical Engineering, University of Thessaly
Fourth Examiner	Dr. Georgios K. Saharidis Assistant Proffesor, Department of Mechanical Engineering, University of Thessaly
Fifth Examiner	Dr. Michael Doumpos Associate Professor, School of Production Engineering and Management, Technical University of Crete
Sixth Examiner	Dr. Athanasios Mygdalas Chaired Professor, Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Sweden
Seventh Examiner	Dr Christos Tarantilis Professor, Department of Management Science and Technology, Athens University of Economics

Acknowledgements

First and foremost, I would like to thank my dissertation supervisor, Assistant Professor Dr. George Kozanidis, for his valuable help and guidance throughout this work. I am also grateful to Dr. George Liberopoulos and Dr. Athanasios Ziliaskopoulos for the valuable knowledge I acquired from them during my post-graduate studies, as well as to the other members of the examining committee of my dissertation, Dr. George Sacharidis, Dr. Michael Doumpos, Dr. Athanasios Mygdalas and Dr. Christos Tarantilis for the thorough examination of my work. I owe grateful thanks to all my professors who helped me enhance my knowledge, so as to obtain the essential theoretical background for the completion of this demanding dissertation. I am also truly thankful to my parents, Thanasis and Meni, and my sister Aspelina for their unconditional love and endless support. Above all, I thank my wife Maria for her understanding and her encouragement, especially through the last period of my effort during which she gave birth to our son, as well as her mother Konstantina for her constant support. I promise to my son Athanasios that I will be the first to teach him the value of education and critical thinking, and I sincerely wish that this effort will be an exemplary guide, so that he always works hard in life for everything he aspires to achieve.

I dedicate this dissertation to my parents, my wife and my son.

Andreas Gavranis

OPTIMAL ALLOCATION OF FLIGHT AND MAINTENANCE HOURS TO MISSION AIRCRAFT. FORMULATIONS, SOLUTION ALGORITHMS AND COMPUTER IMPLEMENTATIONS

ANDREAS GAVRANIS

University of Thessaly, Department of Mechanical Engineering, 2017

Dissertation Advisor: Dr. George Kozanidis, Assistant Professor in Optimization Methods of Production/Service Systems

Abstract

This dissertation addresses the Flight and Maintenance Planning (FMP) problem, i.e., the problem of deciding which available aircraft to fly and for how long, and which grounded aircraft to perform maintenance operations on in a group of aircraft that comprise a unit. FMP is an important decision making problem arising at the typical operation of the Hellenic Air Force (HAF). The aim is to maximize the fleet availability of the unit, while also ensuring that certain flight and maintenance requirements are satisfied.

We develop several optimization models for the formulation of the FMP problem, which accommodate various objective functions as well as constraint sets. These models handle small problem instances effectively, but tend to be computationally inefficient for larger problems, such as the ones that arise in practice. With this in mind, we first develop heuristic approaches which can provide near optimal solutions in insignificant solution times. Due to the fact that these approaches often generate solutions which are far from the optimum, we go on to develop exact solution algorithms for the FMP Problem, which are capable of identifying the exact optimal solution of considerably large realistic problems in reasonable computational times. The first algorithm that we develop handles the single period version of the problem, whereas the second one handles the multi-period one.

A crucial difficulty that often arises in practice relates to the fact that the fleet availability of the solutions provided by the aforementioned methodologies often exhibits significant variability. In order to handle this difficulty, we develop a multi-objective FMP model next, which includes an additional objective minimizing the variability of the fleet availability. For this model, we develop two exact solution methods, which are capable of identifying the entire frontier of non-dominated solutions.

In order to test the performance of the proposed optimization models, we used the commercial optimization solvers IBM ILOG CPLEX and LINGO; for the development of the specialized solution algorithms, we used the C/C++ programming language. The experimental results that we present demonstrate the high efficiency of the proposed solution methodologies on both randomly generated as well as on realistic problem instances, as compared to the traditional approaches that can be used alternatively for the solution of the problem under study.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

Before delving into the mathematical content of this dissertation, we first provide some context and motivation. This dissertation grew out of research in the area of optimization. Optimization deals with the development and application of mathematical programming techniques for the solution of complex decision-making problems. Evidence of vigorous research activity within this field is easy to document. This introduction gives an overview of the motivation for dealing with the specific problem under consideration, states the dissertation's main contributions, and provides a guide through the chapters to follow.

1.1 Motivation and background

The Air Force and the commercial airline industry have several similarities, but also exhibit significant differences. Safety is the most important factor in both industries; however, while maximization of profit is naturally the overall objective in the commercial airline industry, maximization of the readiness to respond to external threats is the main objective in the Air Force. Therefore, military aircraft operational problems should generally be treated differently than traditional problems arising in the commercial airline industry.

A significant part of the total operational budget of a fleet is spent for maintaining the aircraft that comprise it. In the commercial airline industry, there are four different levels of maintenance which differ from each other in philosophy, duration, and frequency of occurrence. They are:

- **Type "A" Check:** This check is performed every 65-100 flight hours or once a week. It includes inspection of all major components and systems of the aircraft, such as the landing system, the engines, and the control surfaces.
- **Type "B" Check:** This check is performed every 300-600 flight hours and includes lubrication of all moving parts and thorough optical testing of several components, such as the rear wing and the slope surfaces. Both Type A and B checks are usually performed overnight so that the aircraft becomes operational in the next morning. If the necessary equipment is available, Type "A" and "B" checks are usually performed on site, at the base location of the aircraft.
- **Type "C" and Type "D" Checks:** These are more costly and time consuming checks performed every one and four years, respectively, which require grounding of the aircraft for several weeks. Type "C" and "D" checks are only performed in special facilities that have the necessary equipment and know-how.

Military aircraft are usually categorized according to their type or the mission that they can accomplish. Typical missions are pilot training, recognition and repulsion of enemy aircraft, bombardment, etc. The safety standards of military aircraft are usually prescribed by their manufacturer; due to the fact that there are a few such manufacturers worldwide, the standards used by Air Force organizations of different countries are often similar. Each aircraft must be grounded for a routine maintenance check as soon as it completes a certain number of flight hours since its last maintenance check. There are also restrictions regarding calendar time (as opposed to flight time) and number of takeoffs, but they are rarely used in practice, because the flight time restrictions usually apply sooner.

In the current dissertation, we address the problem of the joint flight and maintenance planning (FMP) of military aircraft. The FMP problem poses the question of which available aircraft should fly and for how long, and which grounded aircraft should perform maintenance operations, in a group of aircraft that comprise a combat unit. The objective is to achieve maximum availability of the unit over the planning horizon. The FMP problem is a very important decision making problem in the Air Force. Due to the fact that it involves both operations as well as maintenance related decisions, we treat it as a unified operational problem.

The FMP problem arises as an important decision making problem in the typical operation of the HAF. The HAF is primarily responsible for Greece’s national air defence. It is split into four Divisions: Division of Tactical Air Force, Division of Air Support, Division of Air Training, and a fourth division, responsible for other units and services. All units responsible for air operations and missions belong to the Division of Tactical Air Force. Further down the organizational structure of the Division of Tactical Air Force, we find the Combat Wings, which are subdivided into squadrons. The HAF is supported by a three-level maintenance program as follows:

- **1st level maintenance (organizational level):** This check is performed on site and includes inspection, repair, and parts replacement.
- **2nd level maintenance (intermediate level):** This check is performed on site and includes more thorough inspection, repair, and parts replacement than the 1st level maintenance.
- **3rd level maintenance - Manufacturer's maintenance (depot level):** This check is performed in special facilities by specially trained professionals. It includes more thorough repair and parts replacement than the other two levels.

The key challenges faced by military managers around the world are readiness, affordability and increased operational workload. At the same time, the associated budgets become increasingly tighter. This implies that existing aerospace platforms and systems must remain in service much longer than originally expected. The B-52 Stratofortress, for example, which entered the U.S. Air Force inventory in 1954, has served in recent operations "Enduring Freedom" and "Iraqi Freedom" and is expected to continue flying until 2040, after being upgraded between 2013 and 2015.

High operational workload cannot be handled easily by aging fleets, however. Maintenance costs tend to rise, fleet availability decreases, and obtaining out-of-production spare parts becomes expensive and difficult. Platforms also need upgrades to keep them relevant in today's integrated battlespace. It is estimated that, while the development and production of a military aircraft system make up only about 30 percent of a government's total ownership cost, the overwhelming 70 percent of total cost regards sustainment and support functions, such as program planning and data management, training, developing and updating technical manuals, purchasing and managing spare parts and support equipment, and carrying out maintenance, modifications, upgrades and other aging aircraft initiatives.

While the importance of the maintenance functions of fleet organizations has received considerable attention, its linkage to the operations functions has been overlooked. Traditionally, the maintenance functions have been considered separately from the operations functions. One of the main reasons for this separation has been the difficulty in information exchange and the lack of communication between operations and maintenance. As a result, these two functions often appear to operate competitively, although their ultimate aim is common. The vast progress of real time information management systems during recent years has made it possible to look into the entire fleet management organization as an integral system and optimize all its major parts towards the primary mission.

Even though our work is carried out within the context of a military application, our model can be applied to several non-military applications, such as planning for fire-fighting aircraft, rescue choppers, etc. The lack of effective synchronization of flight and maintenance operations in these applications, too, may have devastating results. Such was the case, for example, in the week of August 21-27, 2006, when an immense forest fire in the region of Chalkidiki, Greece, burned more than 13,000 acres of virgin fir and pine forest and olive groves, as well as tens of homes, tourism infrastructure, livestock and agricultural installations and machinery. At the peak of the blaze on the second day, the fire had extended over a 20 mile front. Government officials admitted that the low number of fire-fighting planes that initially responded to the blaze was due to three aircraft experiencing mechanical problems and another six temporarily grounded for regular maintenance (ANA, 2006).

1.2 Dissertation contributions

The main contribution of the present work lies in the development of various mathematical optimization models for the FMP problem along with the specialized algorithms that facilitate their efficient solution. These operations research tools comprise a useful toolset that the aviation commanders and the maintenance managers can utilize to address many aspects of the FMP problem effectively. We also study interesting variants of the problem under consideration, and we illustrate how the proposed methodology can be modified in order to accommodate them. The extensive computational results that we present demonstrate the performance of the proposed solution methodologies, and the impact that several key parameters have on their behavior.

The beginning part of the present research was based on the study of three existing mathematical optimization models for the FMP problem (Kozanidis, 2009; Kozanidis et al., 2010). The first two of these models are single-objective mixed integer linear programs incorporating alternative definitions of the unit availability, while the third one is a multi-objective mixed integer linear program accommodating each of these definitions as a separate objective. The latter of these models seems to exhibit wider applicability and produce, in general, solutions of higher quality. The computational effort that these models need in order to reach the optimal solution increases rapidly with problem size. As a result, their applicability on large scale problem instances such as the ones that arise in practice is quite limited.

With this in mind, we develop two heuristic solution procedures first for solving the FMP problem. The first one utilizes a technique which is widely used in an ad-hoc manner for the production of aircraft flight and maintenance plans in many Air Force organizations worldwide. This technique is based on a practical "sliding scale scheduling" or "aircraft flowchart" graphical tool for scheduling aircraft for phase/periodic inspection and deciding which aircraft should fly certain missions. The second heuristic procedure works by decomposing a large problem into smaller sub-problems and solving each of these subproblems separately. We develop the theoretical background on which these heuristics are based, we provide in detail the algorithmic steps required for their implementation, and we analyze their worst-case computational complexity. We also present computational results illustrating their computational performance on random problem instances, and we evaluate the quality of the solutions that they produce. The size and parameter values of some of these instances are quite realistic, making it possible to infer the performance of the heuristics on real world problem instances. Our computational results demonstrate that, under careful consideration, these heuristics can handle quite large FMP instances effectively, yielding satisfactory solutions in insignificant solution times.

Our computational experience suggests that an effective FMP model should ideally be able to provide solutions whose fleet availability exhibits low variability. This is mainly due to the fact that, since the FMP model is considered in subsequent rolling horizons in practice, the transition into the next planning horizon should always be as smooth as possible. With this in mind, we develop a single-period optimization model that establishes a balanced allocation of the flight load and the maintenance capacity to the individual aircraft of the unit, so that its long term availability is kept at a high and steady level. This model is a mixed integer nonlinear program, which minimizes a least squares index expressing the total deviation of the individual aircraft flight and maintenance times from their corresponding target values. Utilizing this model's special structure and properties, we develop an exact algorithm for obtaining its optimal solution. We analyze the computational complexity of this algorithm, and we present computational results demonstrating that its performance is superior to that of a commercial optimization package that can be utilized alternatively to this end.

Next, we consider the multi-period FMP problem. In order to overcome the excessive computational requirements of exact optimization models and the inferior quality of the solutions produced by heuristic techniques, we develop an exact solution algorithm for this problem. Exploiting the problem's special structure, this algorithm is capable of identifying the optimal solution of considerably large realistic problems in reasonable computational times. This is achieved by solving suitable relaxations of the original problem and utilizing valid cuts which guide the search towards the optimal solution. The extensive experimental results that we present demonstrate that the algorithm's performance on realistic problems is superior to that of two popular commercial optimization software packages; on the other hand, they show that the opposite is true for a class of problems with special characteristics deviating considerably from those of realistic problems. The important conclusion is that the proposed algorithm, complemented by generic optimization software, can handle effectively a large variety of FMP problem instances.

We conclude this dissertation with the development of a mixed integer programming model, which, besides the typical objective maximizing the fleet availability, also includes an additional objective minimizing its variability. Motivated by the fact that the application of the typical ε-constraint reduced feasible region approach to this biobjective model exhibits substantial computational difficulties, we also develop two specialized solution methodologies for this problem. Both methodologies identify the entire frontier of non-dominated solutions, utilizing suitable relaxations of the original model and exploiting the fact that the domain comprising possible fleet availability values is a discrete set. The first one disaggregates the original FMP model into smaller sub-problems whose solution is attained much more efficiently due to their reduced size. The second one is a variant of the ε-constraint method, applied to a suitable relaxation rather than the original FMP model. We present extensive computational results which assess the efficiency of the proposed solution methodologies and demonstrate that their performance is significantly superior to that of the typical ε-constraint method.

1.3 Structure of the dissertation

The present dissertation presents original research, part of which has already been published in scientific journals and is reprinted with permission from the publishers[1]. Its remainder is organized in seven chapters and three appendices, as follows:

In Chapter 2 we review the related literature, while in Chapter 3 we present a detailed definition of the problem under consideration and we elaborate on various issues related to the development of accurate FMP optimization models. Chapter 4 presents the heuristic approaches that we developed for the solution of the FMP problem. Chapters 5 and 6 present mathematical models and associated solution algorithms for the single and the multi-period, respectively, version of the FMP problem.

Chapter 7 presents mixed integer programming models and associated solution algorithms for the version of the problem in which the minimization of the fleet variability is also incorporated as an additional objective. Chapter 8 summarizes the findings of this dissertation and points to fruitful directions for future research. Appendices A and B contain proofs to a key proposition and a key lemma utilized in Chapters 4 and 5, respectively. Appendix C lists the journal and conference publications that have resulted from the present dissertation to date. Finally, Appendix D contains a glossary of dissertation terms and acronyms.

[1] ©Wiley, ©Springer, ©Elsevier

Chapter 2 Literature Review

2.1 Introduction

Numerous problems dealing with the optimization of aircraft operations have been investigated in the past. In this chapter, we review the related literature, focusing mostly on works that address military related applications. First, we review works in the general field of military aircraft maintenance. Then, we turn our attention to works that employ special purpose techniques in order to deal with military aircraft organizational level maintenance activities and mission assignments. We conclude with a review of the published papers which are more closely related to the problem under consideration, i.e., papers which deal with the problem of scheduling military aircraft for intermediate level phase maintenance inspections and mission assignments.

2.2 General military aircraft maintenance works

The increasing importance of effective military aircraft maintenance was recently recognized by the Operations Research and Management Science community (Horner 2006). The 2006 Franz Edelman INFORMS Award for outstanding operations research and management science practice was bestowed on Warner Robins Air Logistics Center (WR-ALC). WR-ALC, located in Georgia, U.S., is responsible for the repair, modification and overhaul of various mission aircraft of the U.S. Air Force, such as the F-15 Eagle and Strike Eagle, the C-130 Hercules models, the C-5 Galaxy, the C-17 Globemaster III, as well as their respective avionics system components. Working with Realization Technologies and faculty from the University of Tennessee, WR-ALC used an operations research technique called Critical Chain to reduce the number of C-5 aircraft undergoing repair and overhaul in the depot from twelve to seven in just eight months. As a direct consequence, the time required to repair and overhaul the C-5 aircraft was reduced by 33%.

Abrahao and Gualda (2006) present the results of a doctoral work which addresses the problem of preventive maintenance scheduling of a fleet of vehicles. Besides developing an optimization model formulation for this problem, the authors develop and test several ant colony based solution approaches considering various instances of the maintenance scheduling problem. They also illustrate the application of the proposed methodology for scheduling the preventive maintenance of an aircraft fleet belonging to the Brazilian Air Force.

Steiner (2006) develops a novel heuristic approach for preventive maintenance scheduling in the Swiss Air Force, which sets up an initial maintenance plan and then employs a heuristic algorithm consolidating the maintenance tasks in order to optimize it. By shifting usage-based and calendar-based maintenance activities in order to realize mergers, the proposed methodology succeeds in minimizing the total maintenance downtime, which, in turn, has a direct positive effect on aircraft availability.

Hahn and Newman (2008) develop a mixed integer linear programming model for scheduling the deployment and maintenance of the United States Coast Guard HH60J helicopters. The proposed model schedules the maintenance of each helicopter based on its flight hours, and decides when it should conduct operations either at home base or at one of two alternative deployment sites. The schedule development considers different maintenance types, as well as the maintenance capacity and various operational requirements such as the number of helicopters simultaneously patrolling a deployment site.

In the context of stochastic approaches, a group of researchers from the Systems Analysis Laboratory of the Helsinki University develop simulation models for the maintenance and repair of a fleet of Bae Hawk Mk51 aircraft, both for the case of normal operation as well as for the case of deployment under crisis situations. The related research has been published in a series of conference papers (Raivio et al., 2001; Mattila et al. 2003, 2008; Mattila and Virtanen 2005, 2006). In a related work, Mattila (2007) considers the assignment of aircraft to flight missions as a markov decision process over a finite time horizon, and develops a methodology utilizing the average aircraft availability as the optimization objective. The problem of finding an efficient assignment policy is solved using a reinforcement learning approach called Q-learning. The performance of the Q-learning approach is compared to a set of heuristic assignment rules using problem instances that involve a variable number of aircraft and various types of periodic maintenance.

Mattila and Virtanen (2014) address the problem of scheduling maintenance for a fleet of fighter aircraft and develop a multi-objective approach based on discrete-event simulation and simulated annealing for the generation of non-dominated solutions. They also develop a multi-attribute decision analysis model to support the maintenance decision maker in selecting the preferred non-dominated solution. The authors choose a different set of objectives than those considered in the present work, which focuses on maximizing the average aircraft availability and minimizing the average deviation between the target and the actual starting times of the maintenance activities. The work of Mattila and Virtanen differs from the present one in that it proposes a metaheuristic solution approach, and in that it considers a stochastic model which incorporates uncertain durations for the aircraft activities as well as the possibility of unexpected failures.

2.3 Works on scheduling organizational level maintenance

Several published works address the problem of assigning a group of available aircraft to missions and organizational level repair activities, so as to establish a high level of unit readiness. This is the case with the work of Yeung et al. (2007), who develop a model-based methodology for mission assignment and maintenance scheduling of systems with multiple states. The authors utilize heuristics and simulation to solve the model, and illustrate its application on a hypothetical scenario of a fleet of aircraft.

Safaei et al. (2011) develop a mixed integer optimization model to formulate the problem of workforce-constrained maintenance scheduling for a fleet of military aircraft. The goal is to maximize the aircraft that can be assigned to missions under maintenance scheduling and workforce availability constraints. The model utilizes a network flow structure in order to simulate the flow of aircraft between missions, the hangar and the repair shop, and is solved with generic optimization software.

In a recent related work, Bajestani and Beck (2013) address a dynamic repair shop scheduling problem that takes into consideration flight requirements, aircraft failures, as well as maintenance related capacity constraints. The goal is to assign aircraft to flights and schedule repair jobs, so as to maximize the coverage of the unit flight requirements. The authors accommodate the stochasticity that the problem exhibits by decomposing it into smaller static sub-problems, and propose several alternative solution methodologies, including mixed integer programming, constraint programming, logic-based Benders decomposition, and heuristics.

2.4 Works on intermediate level scheduled-phase maintenance

Although FMP is an important decision making problem encountered in several diversified areas, the relevant published research is rather limited. Sgaslik (1994) introduces a decision support system for maintenance planning and mission assignment of a helicopter fleet that partitions the master problem into two subproblems which are solved separately. The first subproblem is called the Yearly Planning Model (YPM). The YPM assigns helicopters to inspections and exercises, while also providing their maintenance schedule and their flight hour distribution. The second model is called the Short Term Planning Model (STPM). The STPM takes as input the maintenance schedule produced by the YPM and returns the helicopters' mission assignments. The author develops two elastic mixed integer programs to formulate these two sub-problems and solves them using standard optimization software. The YPM minimizes the cost associated with the violation of some of the problem's constraints (e.g., those referring to the required flight time, the maintenance capacity and the flight time of each individual aircraft), while also maintaining a given lower bound on the fleet availability.

Pippin (1998) develops a MILP and a quadratic program for the FMP problem, which try to establish a flight hour allocation that ensures a steady-state sequence of aircraft into phase maintenance. Both these models minimize the cost associated with the deviations of the individual aircraft residual flight times from their diagonal line target values, but neither of them incorporates the apparent difficulties introduced by the maintenance aspect of the problem.

Kozanidis (2009) proposes a multi-objective MILP model for the FMP problem that maximizes the minimum aircraft and flight time availability of the wing and of the squadrons that comprise it. The proposed methodology utilizes the weighted sums approach (Geoffrion, 1968; Steuer, 1986) for solving the problem, which cannot, in general, provide the entire non-dominated set. Kozanidis et al. (2010) develop a single objective optimization model that maximizes wing aircraft availability while imposing a lower bound on the number of available aircraft of each squadron over all periods and a lower bound on the average residual flight time of each available aircraft.

The U.S. Department of the Army has released a Field Manual on Army Aviation Maintenance, which describes a practical “sliding scale scheduling” or “aircraft flowchart” graphical tool for scheduling aircraft for phase/periodic inspection and deciding which aircraft should fly in certain missions (US DoA, 2000). Rosenzweig et al. (2010) develop a MILP to formulate the sliding scale method for deciding the aircraft flight times. This model minimizes the penalty associated with the deviation of the aircraft flight times from their diagonal target values, but does not consider the maintenance requirements and the impact that they can have on the fleet availability of the unit. The authors solve the model with generic optimization software and illustrate its application on a small fleet of training aircraft.

Cho (2011) develops a MILP to model the FMP problem. The proposed formulation generates a daily flight and maintenance plan that distributes the maintenance workload evenly across the planning horizon. The main difference that this model exhibits with respect to the one that we address in the current work is that it uses different definitions for the objective function and for the flight requirements of the unit. With respect to the former, that model minimizes the maximum number of aircraft in maintenance at any given time in order to smoothen the variability of the maintenance demand over time. With respect to the latter, it translates the original flight load requirements into specific flight assignments, which are successively assigned to the aircraft of the unit. The author also considers a two-stage formulation that disaggregates the problem in order to determine the flight and maintenance decisions separately. All the decisions pertaining to the flight or the maintenance schedule are made in the first stage, while the remaining ones are determined in the second one. Both the single and the two stage models are solved with generic optimization software, although a discussion that proposes equivalent alternative formulations and potential heuristic solution approaches is also included.

Based on the work of Kozanidis and Skipis (2006), Verhoeff et al. (2015) develop a flight and maintenance planning optimization model that incorporates the aspects of availability, serviceability and sustainability for the RNAF CH47D Chinook helicopter fleet. The proposed model formulation maximizes the minimum scheduled sustainability over the planning horizon, while also ensuring that the variability of the residual flight time availability remains relatively low.

Chapter 3 FMP Problem Definition and Model Development Considerations

3.1 Introduction

Having presented a thorough literature review, in this chapter we address model development issues related to the FMP problem. Specifically, we present the detailed problem definition, we elaborate on alternative FMP objective function choices, and we discuss various issues related to the development of accurate FMP optimization models. We also provide a basic overview of a common ad-hoc approach that has been utilized to address the FMP problem.

3.2 Problem definition

The FMP problem arises as a routine operations management problem in a typical aircraft unit (typically, a combat wing) of the HAF. Depending on the particular context, such a unit may consist of several squadrons, each of which serves as the base for several aircraft types. When this is the case, we often use the term "wing" to refer collectively to all the squadrons forming the unit together.

In order to retain a high level of unit readiness, at the beginning of each planning horizon the unit command issues suitable flight requirements, which are also referred to as *flight load*. These requirements determine the total time that the aircraft of the unit should fly in each corresponding time period, and only small deviations are permitted from them. Separate requirements are issued for each aircraft type, because different aircraft types have different flight capabilities and maintenance needs. For this reason, the optimization models that we develop for addressing the FMP problem are suitable for use on a specific aircraft type. Of course, each of these models can be applied repeatedly in order to issue the plans of several aircraft types, if more than one aircraft types are present.

For each individual aircraft, we define its *residual flight time* as the total remaining time that this aircraft can fly until it has to undergo a maintenance check. This time is also referred to as *"bank time"* in the related military literature (US DoA, 2000). The residual flight time of an aircraft is positive if and only if this aircraft is available to fly. At any time, the total residual flight time of the unit is equal to the sum of the residual flight times of all the aircraft that belong to this unit. Of course, there exist many possible combinations of individual aircraft residual flight times that can result in the same total residual flight time for the unit. Similarly, we define the *residual maintenance time* of a non-available aircraft as the total remaining time that this aircraft needs in order to complete its maintenance check. The residual maintenance time of an aircraft is positive if and only if this aircraft is undergoing a maintenance check, and is therefore not available to fly.

For the maintenance needs of the unit, there exists a station responsible for providing service to its aircraft. This station has certain space (also referred to as *dock space*) and time capacity capabilities. Given the flight requirements for each time period and the physical constraints that stem from the capacity of the maintenance station, the aim of the FMP problem is to issue a flight and maintenance plan for each individual aircraft, so that the unit's readiness to respond to external threats (*operational readiness*) is maximized. In the military context, the operational readiness of a unit is defined as follows by the North Atlantic Treaty Organization (NATO, 2015): "The capability of a unit/formation, ship, weapon system or equipment to perform the missions or functions for which it is organized or designed. May be used in a general sense or to express a level or degree of readiness."

In accordance with the above definition, the readiness of a unit to respond to external threats is defined as the capability of the unit to perform the assigned flight missions. This capability is expressed in terms of the total number of aircraft that are available to fly (aircraft availability) and in terms of the total residual flight time of all available aircraft (residual flight time availability).

The FMP problem refers mainly to the intermediate level scheduled maintenance, also called *phased maintenance*, which is a time consuming activity that may lead to extended grounding of the aircraft, and, as a consequence, affect adversely the unit fleet availability. It is a very important decision making problem arising in the operation of numerous types of fleets, involving military or fire-fighting aircraft, rescue choppers, etc.

3.3 Aircraft flowchart

A common empirical approach for addressing the FMP problem involves the utilization, in an ad-hoc manner, of a 2-dimensional graphical tool called the *aircraft flowchart*, shown in Figure 3.1.

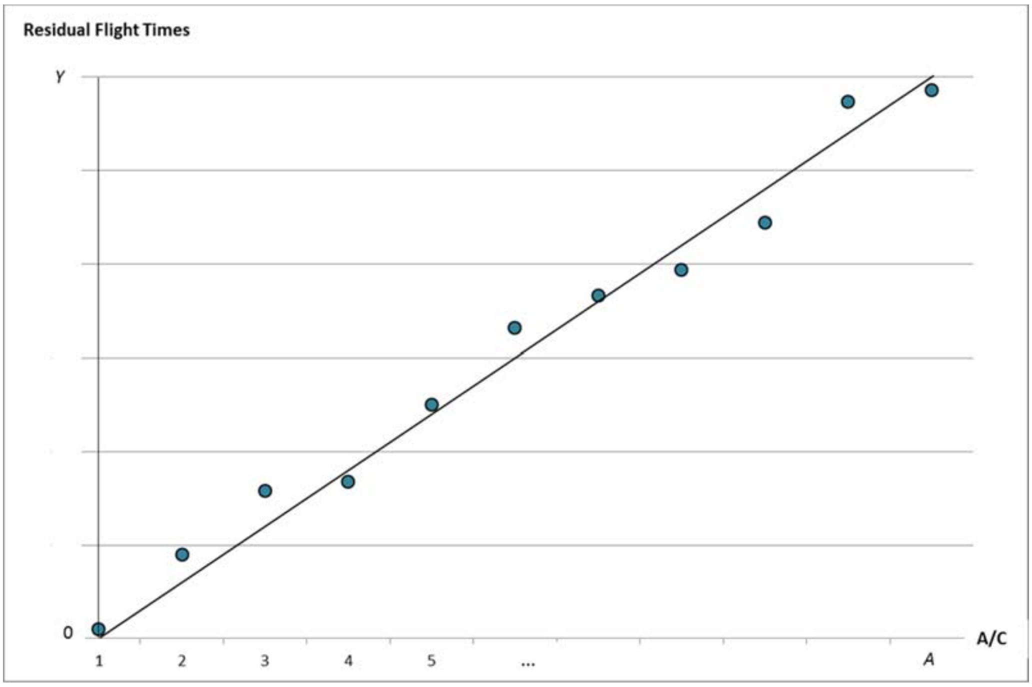

Figure 3.1: Visual representation of the fleet availability with the aircraft flowchart

The vertical axis of this graph represents residual flight time, while the horizontal axis represents the indices of the available aircraft in non-decreasing order of their residual flight times, 1 being the index of the aircraft with the smallest, and *A* being the index of the aircraft with the largest residual flight time, where *A* is the total number of available aircraft. Consider the line segment (also called the *diagonal*) that connects the origin with the point with coordinates (*A*, *Y*), where *Y* is the total flight time of an aircraft between two maintenance inspections, also called *phase interval*. By mapping each aircraft on this graph, we can visualize the unit's fleet availability.

To describe the smoothness of the distribution of the total residual flight time among the available aircraft, a *total deviation index* is used. This index is equal to the sum of squares of the vertical distances (deviations) of the points mapping the residual flight times of the individual aircraft from their corresponding target values on the diagonal. The smaller this sum is, the smoother the distribution of the total residual flight time. Ideally, the total deviation index is equal to 0, when every point lies on the diagonal, as shown in Figure 3.2. When issuing the flight plan of each individual aircraft, the user is advised to keep each point as close to the diagonal as possible, in order to minimize the value of the total deviation index.

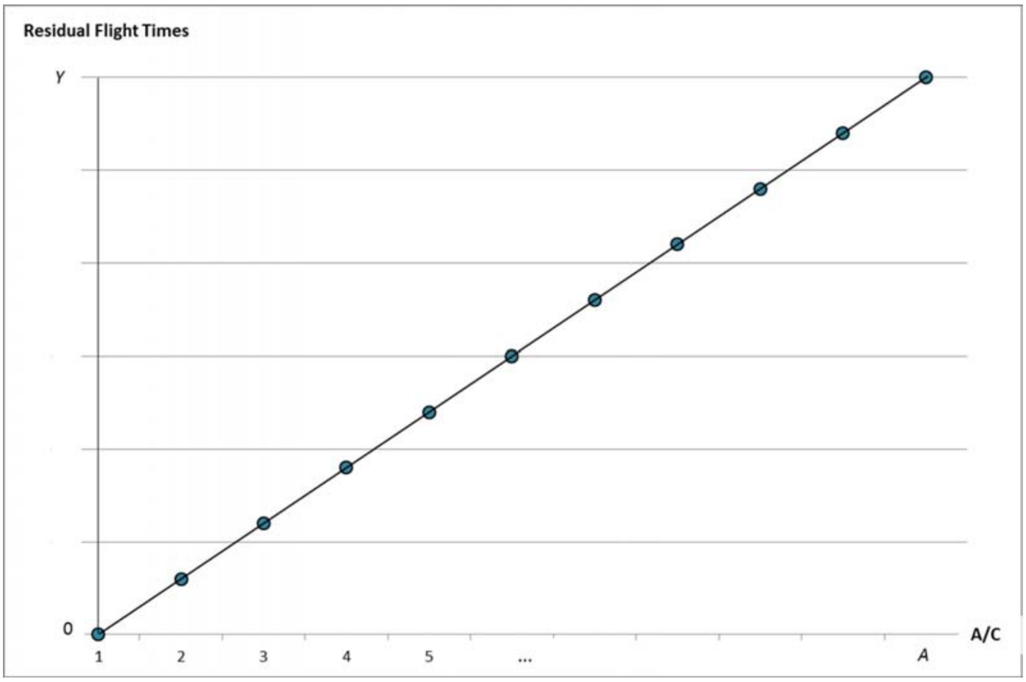

Figure 3.2: Ideal fleet availability distribution among the aircraft

In current practice, the aircraft flowchart is at best used as a graphical device by the aviation commanders and the maintenance managers responsible for issuing the flight and maintenance plans. For example, in an aviation maintenance manual of the U.S. Army where this flowchart is described (US DoA, 2000), the user is simply advised to utilize the flowchart by "flying the aircraft that are above the diagonal to get them down to the line" and "holding the aircraft that are below the diagonal to bring them up to the line". No particular instructions are given on how this can be implemented effectively. Clearly, this procedure is highly subjective and dependent on numerous minor decisions made by the user.

3.4 Alternative FMP objective functions

One of the most crucial decisions that need to be made towards the development of FMP optimization models regards the choice of the objective function. Maximizing the readiness to respond to external threats is the most appropriate measure of effectiveness for this application. As already explained, in the military context the readiness of a unit to respond to external threats is typically expressed as the total number of aircraft that are available to fly (aircraft availability) and as the total residual flight time of all available aircraft (residual flight time availability). Although a certain degree of synergy between the two exists, optimizing one of them may, in some cases, have an adverse effect on the optimization of the other. Moreover, knowing one of the two gives no information about the way that the other is distributed across the individual aircraft.

This situation is highlighted in the example depicted in Figure 3.3, in which there is a large number of available aircraft, but both the unit residual flight time availability, as well as the separation between phases (periodic inspections), are particularly low. The label on top of each point in this figure reveals the residual flight time of the corresponding aircraft. This situation is highly problematic; many aircraft will soon need to enter the station for phase maintenance, but only a small number of them will be able to do so due to the limited space capacity of the maintenance station. As a consequence, the imminent flight load requirements of the unit cannot be met adequately.

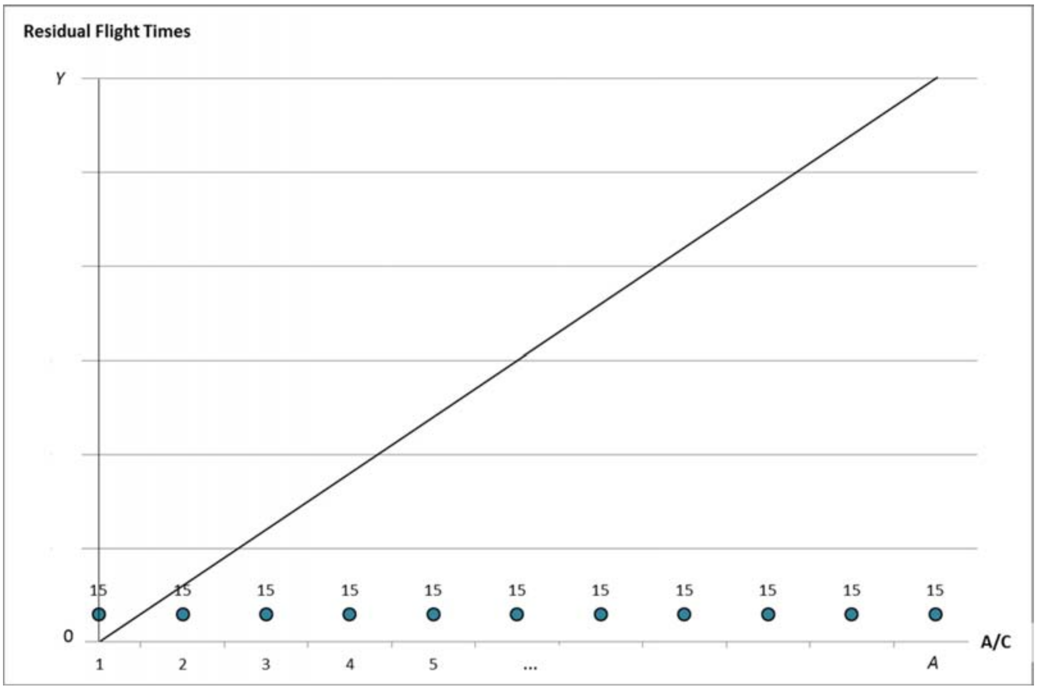

Figure 3.3: An example of high aircraft availability and low residual flight time availability

An alternative situation is highlighted in the example depicted in Figure 3.4, in which the unit residual flight time availability is high, but the number of available aircraft (aircraft availability) is particularly low. In addition there is a gap in the separation between phases (periodic inspections), similar to that of the previous example. This situation is also problematic, because the number of available aircraft is not adequate to perform the necessary missions.

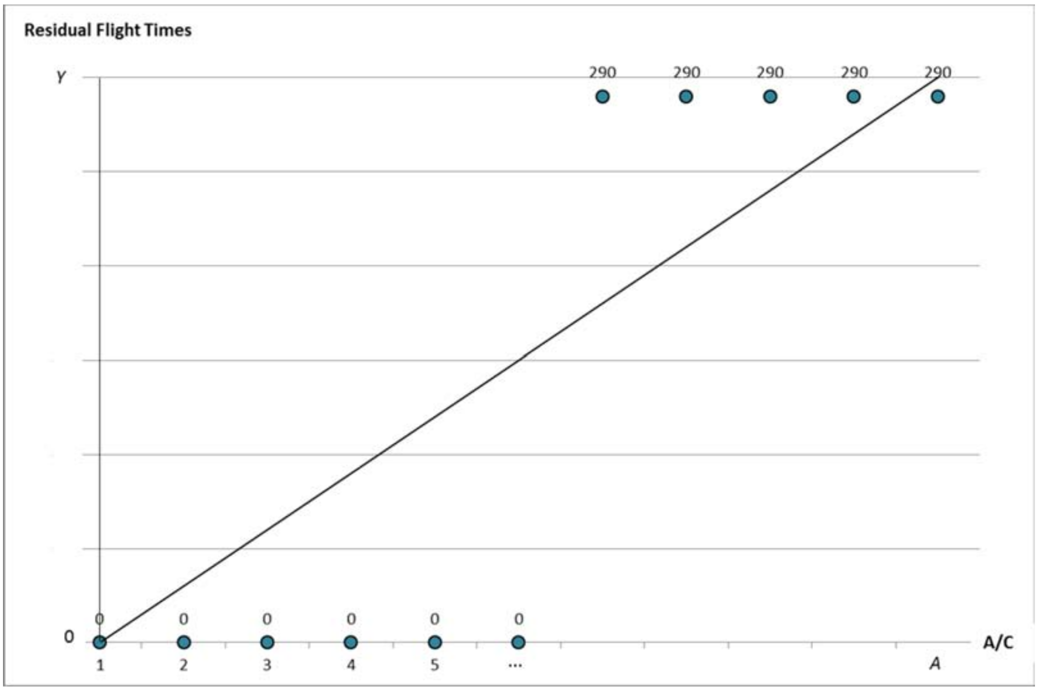

Figure 3.4: An example of high residual flight time availability and low aircraft availability

Another related important decision concerns the question of whether the minimum or the total fleet availability over a given planning horizon should be used as the problem objective. In the former case, the focus is on finding the highest availability level that can be ensured for each time period of the planning horizon, whereas in the latter one, the focus is on finding the highest availability level that can be attained cumulatively over all time periods of the planning horizon, independently of how this varies between individual ones. A common strategy that is often used in practice is to maximize the total fleet availability, while also imposing an acceptable lower bound on the fleet availability of each individual time period.

3.5 FMP model formulation considerations

Besides the objective function selection, several other important decisions are involved in the process of developing an accurate FMP optimization model. In practice, these decisions should be made based on the specific characteristics and requirements of each particular application. Furthermore, a systematic study that evaluates and compares different choices may also be necessary in order to choose the most appropriate model. In an effort to address a wide range of different problem characteristics, we develop various FMP optimization model variants in this work. The various design choices the user is faced with are discussed next and involve the length of the planning horizon, the structure of the fleet unit, the satisfaction of the flight load and the number of objective functions.

In practice, the flight load requirements refer to single-month time periods and are typically issued over a planning horizon of 6 monthly periods. Several FMP optimization models focus on maximizing the fleet availability within each individual planning horizon in isolation, without taking into consideration the fact that, since FMP is an on-going problem repeatedly solved in successive horizons, the transition into the next planning horizon must also be as smooth as possible. This clearly results in plans which, although being optimal within each individual planning horizon, do not exhibit certain desirable long-term characteristics, such as low variability. An inevitable side effect of this behavior is the fact that the exact length of each planning horizon strongly affects the pattern of the fleet availability; the longer this length, the lower the associated variability is expected to be. Still, however, since the application of the model in subsequent rolling horizons is inevitable, the selection of the planning horizon length is an important modeling choice.

As already mentioned, an aircraft unit (wing) may consist of several squadrons. While there is only a single maintenance station responsible for the inspection of all aircraft, each squadron is assigned separate flight load requirements. Wing officials are responsible for monitoring the fleet availability of the wing, whereas squadron officials are responsible for monitoring the fleet availability of the corresponding squadron. The incorporation of multi-squadron units adds a strong combinatorial component to the problem, since distinct cases depending on which squadron an aircraft that enters or exits the maintenance station belongs to need to be distinguished. The decision on whether the model should accommodate distinct squadrons depends on the scheduling needs of the aviation commanders and the maintenance managers and constitutes another important modeling choice.

In many practical cases, the actual problem definition calls for satisfaction of the flight load requirements within some predefined tolerance, instead of their strict satisfaction. For example, a maximum deviation of 5% from the target value of the flight load may be acceptable for each time period of the planning horizon. The main effect of this is that it makes the total flight time of each time period a decision variable instead of a known parameter.

Finally, single objective models only optimize a specific performance measure of a decision making problem. However, in case that multiple performance measures need to be optimized simultaneously, a corresponding multi-objective optimization model needs to be developed instead. When this is the case, the adopted objective functions are usually conflicting, necessitating the search for a suitable compromise between the alternative objective function levels.

Chapter 4 Heuristic Solution Techniques

4.1 Introduction

Simple heuristic techniques used in practice to solve the FMP problem, such as the aircraft flowchart tool (US DoA, 2000), often perform poorly generating solutions that are far from the optimum. On the other hand, the more sophisticated mathematical models that have been developed to handle this problem (e.g., Kozanidis, 2009; Kozanidis et al., 2010) solve small problems effectively, but tend to be computationally inefficient for larger problems that often arise in practice. In this chapter, we consider a multi-objective optimization model for the multi-period variant of the FMP problem, and we develop two heuristic approaches for solving it.

The two heuristic algorithms that we develop have been roughly sketched in the earlier works of Kozanidis (2009) and Kozanidis et al. (2010). In this chapter, we extend these two works by: i) developing the theoretical background on which the proposed heuristics are based, ii) providing in detail the algorithmic steps required for the implementation of these heuristics, iii) analyzing the worst-case computational complexity of these heuristics, iv) presenting computational results demonstrating the computational performance of these heuristics on random problem instances, and, v) evaluating the quality of the solutions that these heuristics produce.

4.2 Multi-objective multi-period FMP model formulation

Having provided the FMP problem definition in the previous chapter, we present next a mathematical model formulation that has been developed for the case in which the aircraft unit (also termed wing) is divided into distinct squadrons. This formulation adopts the following mathematical notation:

Sets:

M : set of squadrons, indexed by m,

N_m : set of aircraft in squadron m, indexed by n.

Parameters:

T : length of the planning horizon,

S_{mt} : flight load of squadron m in period t,

B_t : time capacity of the maintenance station in period t,

C : space capacity of the maintenance station,

Y : residual flight time of an aircraft immediately after it exits the maintenance station,

G : residual maintenance time of an aircraft immediately after it enters the maintenance station,

$A1_{mn}$: state (0/1) of aircraft n of squadron m at the first period of the planning horizon,

$Y1_{mn}$: residual flight time of aircraft n of squadron m at the first period of the planning horizon,

$G1_{mn}$: residual maintenance time of aircraft n of squadron m at the first period of the planning horizon,

X_{max} : maximum flight time of an available aircraft in a single time period,

Y_{min} : lower bound on the residual flight time of every available aircraft,

G_{min} : lower bound on the residual maintenance time of every grounded aircraft,

L, U : real numbers denoting the maximum deviation from the target value of the flight load that can be tolerated,

K : a sufficiently large number.

Decision Variables:

z_1 : minimum number of available aircraft of the wing over all periods,

z_2 : minimum number of available aircraft in each squadron over all periods,

z_3 : minimum residual flight time of the wing over all periods,

z_4 : minimum residual flight time of each squadron over all periods,

a_{mnt} : binary decision variable that takes the value 1 if aircraft n of squadron m is available in period t, and 0 otherwise,

y_{mnt} : residual flight time of aircraft n of squadron m at the beginning of period t,

x_{mnt} : flight time of aircraft n of squadron m in period t,

g_{mnt} : residual maintenance time of aircraft n of squadron m at the beginning of period t,

h_{mnt} : maintenance time of aircraft n of squadron m in period t,

d_{mnt} : binary decision variable that takes the value 1 if aircraft n of squadron m exits the maintenance station at the beginning of period t, and 0 otherwise,

f_{mnt} : binary decision variable that takes the value 1 if aircraft n of squadron m enters the maintenance station at the beginning of period t, and 0 otherwise,

q_t, p_{mnt}, r_{mnt} : auxiliary binary decision variables.

The proposed FMP model (Kozanidis, 2009) is a mixed integer multi-objective linear program with four objectives, formulated as follows:

$$\text{Max } z_1 \tag{4.2.1}$$

$$\text{Max } z_2 \tag{4.2.2}$$

$$\text{Max } z_3 \tag{4.2.3}$$

$$\text{Max } z_4 \tag{4.2.4}$$

$$\text{s.t. } z_1 \le \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} a_{mnt}, \; t = 2,...,T+1 \tag{4.2.5}$$

$$z_2 \le \sum_{n=1}^{|N_m|} a_{mnt}, \; m = 1,...,|M|, \; t = 2,...,T+1 \tag{4.2.6}$$

$$z_3 \le \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} y_{mnt}, \; t = 2,...,T+1 \tag{4.2.7}$$

$$z_4 \le \sum_{n=1}^{|N_m|} y_{mnt}, \; m = 1,...,|M|, \; t = 2,...,T+1 \tag{4.2.8}$$

$$y_{mnt+1} = y_{mnt} - x_{mnt} + Yd_{mnt+1}, \; m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.9}$$

$$d_{mnt+1} \ge a_{mnt+1} - a_{mnt}, \quad m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.10}$$

$$a_{mnt+1} - a_{mnt} + 1.1(1 - d_{mnt+1}) \ge 0.1, \quad m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.11}$$

$$g_{mnt+1} = g_{mnt} - h_{mnt} + Gf_{mnt+1}, \quad m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.12}$$

$$f_{mnt+1} \ge a_{mnt} - a_{mnt+1}, \quad m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.13}$$

$$a_{mnt} - a_{mnt+1} + 1.1(1 - f_{mnt+1}) \ge 0.1, \; m = 1,...,|M|, \; n = 1,\ldots,|N_m|, \; t = 1,..,T \tag{4.2.14}$$

$$LS_{mt} \le \sum_{n=1}^{|N_m|} x_{mnt} \le US_{mt}, \; m = 1,...,|M|, \; t = 1,...,T \tag{4.2.15}$$

$$\sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} h_{mnt} \le B_t, \; t = 1,...,T \tag{4.2.16}$$

$$\sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} (1 - a_{mnt}) \le C, \; t = 2,...,T+1 \tag{4.2.17}$$

$$B_t \leq \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} h_{mnt} + K(1-q_t),\ t=1,...,T \tag{4.2.18}$$

$$\sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} g_{mnt} \leq \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} h_{mnt} + Kq_t,\ t=1,...,T \tag{4.2.19}$$

$$y_{mnt} + Kp_{mnt} \leq K,\ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.20}$$

$$a_{mnt+1} \leq (y_{mnt} - x_{mnt})K + Kp_{mnt},\ \ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.21}$$

$$g_{mnt} + Kr_{mnt} \leq K,\ \ \ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.22}$$

$$1\text{-}a_{mnt+1} \leq (g_{mnt} - h_{mnt})K + Kr_{mnt},\ \ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.23}$$

$$y_{mnt} \leq Ya_{mnt},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,...,T+1 \tag{4.2.24}$$

$$g_{mnt} \leq G(1\text{-}a_{mnt}),\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,...,T+1 \tag{4.2.25}$$

$$x_{mnt} \leq X_{max}a_{mnt},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,...,T \tag{4.2.26}$$

$$y_{mnt} \geq Y_{min}a_{mnt},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,...,T+1 \tag{4.2.27}$$

$$g_{mnt} \geq G_{min}(1 - a_{mnt}),\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,...,T+1 \tag{4.2.28}$$

$$x_{mnt} \leq y_{mnt},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,...,T \tag{4.2.29}$$

$$h_{mnt} \leq g_{mnt},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,...,T \tag{4.2.30}$$

$$a_{mn1} = A1_{mn},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m| \tag{4.2.31}$$

$$y_{mn1} = Y1_{mn},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m| \tag{4.2.32}$$

$$g_{mn1} = G1_{mn},\ \ m=1,\ldots,|M|,\ n=1,\ldots,|N_m| \tag{4.2.33}$$

$$x_{mnt},\ h_{mnt} \geq 0;\ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.34}$$

$$y_{mnt},\ g_{mnt} \geq 0;\ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=2,..,T+1 \tag{4.2.35}$$

$$p_{mnt},\ r_{mnt},\ q_t \text{ binary},\ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{4.2.36}$$

$$a_{mnt},\ d_{mnt}, f_{mnt} \text{ binary},\ m=1,...,|M|,\ n=1,\ldots,|N_m|,\ t=2,..,T+1 \tag{4.2.37}$$

The objective function (4.2.1) maximizes z_1, which, by constraint set (4.2.5), denotes the minimum number of available aircraft of the wing over all periods, while the objective function (4.2.2) maximizes z_2 which, by constraint set (4.2.6) denotes the minimum number of available aircraft of each squadron over all periods. Similarly, the objective functions (4.2.3) and (4.2.4) maximize z_3 and z_4 which, by constraint sets (4.2.7) and (4.2.8), denote the minimum residual flight time of the wing and of each squadron, respectively, over all periods. Although the four objectives do not seem to be in direct conflict with each other, all of them must be included in the model, otherwise the obtained solution may not be satisfactory with respect to the objective that was omitted. Note that, under the weighted sums approach, the 4 objectives are weighted, taking into account the tradeoffs that a decision-maker is willing to make among the objectives. In this study, however, an actual decision-maker was not involved in the analysis. Therefore, the weights of the objectives have been defined to be inversely related to the objectives' scales.

Constraint set (4.2.9) updates the residual flight time of each aircraft at the beginning of the next period, based on its residual flight time at the beginning of the previous period and the time that it flew during that period. Binary variable d_{mnt+1} takes the value 1 only when the corresponding aircraft exits the maintenance station at the beginning of period t+1. In this case, its residual flight time is reset to Y (also referred to as *phase interval*). Similarly, constraint set (4.2.12) updates the residual maintenance time of each aircraft at the beginning of the next period, based on its residual maintenance time at the beginning of the previous period and the time that it received maintenance during that period. Binary variable f_{mnt+1} takes the value 1 only when the corresponding aircraft enters the maintenance station for service at the beginning of period t+1. In this case, its residual maintenance time is reset to G.

Constraint sets (4.2.10), (4.2.11), (4.2.13) and (4.2.14) ensure that variables d_{mnt} and f_{mnt} take appropriate values, based on the values of variables a_{mnt}. More specifically, consider the n^{th} aircraft of the m^{th} squadron. Then, (a_{mnt},a_{mnt+1}) can take any of the values (0,1), (0,0), (1,0) and (1,1) and the dierence $(a_{mnt+1} - a_{mnt})$ is equal to 1, 0, -1 and 0, respectively. Variable d_{mnt+1} should take the value 1 when (a_{mnt},a_{mnt+1}) = (0,1) and this is ensured by constraint set (4.2.10). In any other case, d_{mnt+1} should be equal to 0 and this is ensured by constraint set (4.2.11). Similarly, variable f_{mnt+1} should take the value 1 when (a_{mnt},a_{mnt+1}) = (1,0) and this is ensured by constraint set (4.2.13). In any other case, f_{mnt+1} should be equal to 0 and this is ensured by constraint set (4.2.14).

Constraint set (4.2.15) restricts the flight time of squadron m in period t to the interval $[LS_{mt}, US_{mt}]$ defined by variables L and U, ensuring that the flight load of each squadron and period combination is satisfied. For example, when $L = 0.95$ and $U = 1.05$, a maximum deviation of 5% from the target values of the flight requirements is permitted. Constraint sets (4.2.16) and (4.2.17) ensure that the time and space capacity constraints of the maintenance station will not be violated in any time period. Constraint sets (4.2.18) and (4.2.19) ensure that the maintenance station will not idle whenever there is at least one aircraft waiting for service. With the introduction of the auxiliary binary variables q_t, it is ensured that the total maintenance time provided by the station in period t will be equal to the minimum between the total time capacity of the station, and the total maintenance requirements in this period.

Constraint sets (4.2.20) and (4.2.21) ensure that an aircraft's availability ceases as soon as its residual flight time drops to 0. If $y_{mnt} > 0$, the auxiliary binary decision variable p_{mnt} in constraint

(4.2.20) is forced to 0-value. In this case, constraint (4.2.21) forces a_{mnt+1} to 0-value if $y_{mnt} = x_{mnt}$, since this implies that the residual flight time of this aircraft drops to 0 at the end of period t. Similarly, constraint sets (4.2.22) and (4.2.23) ensure that an aircraft becomes available as soon as its residual maintenance time drops to 0. If $g_{mnt} > 0$, the auxiliary binary decision variable r_{mnt} in constraint (4.2.22) is forced to 0-value. In this case, constraint (4.2.23) forces a_{mnt+1} to 1 if $g_{mnt} = h_{mnt}$, since this implies that the residual maintenance time of this aircraft drops to 0 at the end of period t.

Constraint set (4.2.24) states that the residual flight time of an aircraft cannot exceed Y, and ensures that it will be 0 whenever this aircraft is not available. Similarly, constraint set (4.2.25) states that the residual maintenance time of an aircraft cannot exceed G, and ensures that it will be 0 whenever this aircraft is available. Constraint set (4.2.26) imposes an upper bound on the maximum time that an aircraft can fly during a single time period. Such a restriction is usually present due to technical reasons. Constraint set (4.2.27) imposes a lower bound on the residual flight time of each available aircraft, and constraint (4.2.28) imposes a lower bound on the residual maintenance time of each non-available aircraft. These constraints are introduced to prevent an aircraft from ending up with a negligible but positive residual flight or maintenance time. Constraint set (4.2.29) ensures that the total time that an aircraft flies during a single period does not exceed its residual flight time at the beginning of the same period. Similarly, constraint set (4.2.30) ensures that the total time that the maintenance crew works on an aircraft during a single period does not exceed the residual maintenance time of this aircraft at the beginning of the same period.

Constraint sets (4.2.31), (4.2.32) and (4.2.33) are used to initialize the state of the system at the first period of the planning horizon. When an aircraft exits or enters the maintenance station at the first period of the planning horizon, its residual flight and maintenance times are updated directly; therefore, variables d_{mn1} and f_{mn1} are never used. Finally, constraints (4.2.34), (4.2.35) and (4.2.36), (4.2.37) are the non-negativity and the integrality constraints, respectively.

Instead of the minimum fleet availability of the unit that is maximized in the above formulation, the cumulative fleet availability of the unit may be maximized alternatively. To incorporate this, we also consider two additional objectives for the above model, which maximize the number of available aircraft and the residual flight time of the unit cumulatively over all periods of the planning horizon. Of course, the distinction between wing and squadron availability is redundant in this case. Mathematically, these two objectives are expressed as follows:

$$\text{Max } z_5$$

$$\text{Max } z_6$$

$$\text{s.t. } z_5 \le \sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} \sum_{t=2}^{T+1} a_{mnt}$$

$$z_6 \leq \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|}\sum_{t=2}^{T+1} y_{mnt}$$

The computational effort that the above FMP model requires in order to reach an optimal solution increases fast with problem size (Kozanidis, 2009). As a result, its applicability on large problems is quite limited. This raises the need to develop alternative intelligent approaches in order to address large FMP instances. To this end, in this chapter we develop two heuristic solution procedures, which we call Aircraft Flowchart Heuristic (AFH) and Horizon Splitting Heuristic (HSH), respectively. The above work sketches briefly these two heuristic solution procedures, but does not present the theoretical background, the detailed algorithmic description, the computational complexity analysis and the extensive computational results that we provide next.

4.3 Aircraft Flowchart Heuristic (AFH)

A common empirical approach to the FMP problem involves the utilization, in an ad-hoc manner, of a 2-dimensional graphical tool called the *aircraft flowchart*, presented in Section 3.3. AFH aims to implement this procedure more systematically. We consider two different variants of this heuristic. The first one takes into consideration the squadron each aircraft belongs to, whereas the second one focuses on the wing and treats the aircraft as if they all belong to the same squadron. We call these two variants AFH1 and AFH2, respectively, and we describe them next.

4.3.1 Aircraft Flowchart Heuristic 1 (AFH1)

The application of AFH1 requires a series of decisions in each period of the planning horizon. In order to make these decisions, AFH1 determines a priority order for the unit's squadrons, by computing a *priority index* for each of them. The first such decision regards the time capacity of the maintenance station. Knowing how this capacity will be allocated among the grounded aircraft determines the number of dock spaces that will become available at the end of the current time period. In turn, this has an effect on the production of the flight plan of each available aircraft.

In general, the priority index of each squadron is an indicator of how heavy its anticipated flight load is with respect to its availability. The higher the priority index of a squadron, the higher the priority that is given to the grounded aircraft of this squadron at the maintenance station. In our study, we define the priority index of squadron m at the end of period t ($0 \leq t \leq T$-1) as $S_{mt+1}/\sum_{n=1}^{|N_m|} y_{mnt+1}$. Thus, the higher the flight requirements of a squadron in the next time period with respect to the residual flight time availability of that squadron in the same time period, the higher its priority index is expected to be. Of course, the interested user can come up with additional definitions if necessary, either by modifying this one, or by devising entirely new ones.

In each time period, the maintenance station continues incomplete service which is pending from the previous period and allocates the extra time capacity to the remaining grounded aircraft in the order determined by the priority indices of the squadrons. In order to free dock space and increase the fleet availability of the unit, the station works continuously on the same aircraft until the service of that aircraft is completed. Of course, the service of an aircraft may be spread out over more than one time periods if not enough time capacity exists. Every time an aircraft finishes its maintenance service, the priority index of the squadron it belongs to is updated, and the relative priority order of the squadrons is adjusted accordingly.

The above procedure determines the number of dock spaces that will become available at the beginning of period t+1. These spaces will be occupied by the aircraft that will enter the station for service at the beginning of period t+1. An important decision that must be made subsequently pertains to the order in which the squadrons will be considered for occupying the free dock space. Not surprisingly, we utilize the priority indices to make this decision, too. More specifically, the aircraft of the squadron with the highest priority index is considered for occupying dock space which is available at the maintenance station first, and so on, until either no free dock space exists, or all squadrons have been considered and no other aircraft will be grounded for service at the next time period.

The existence of a free dock space at the maintenance station does not automatically impose the grounding of an available aircraft. Given that it is feasible, the grounding of an aircraft takes place if its service is anticipated to begin in the next time period. This condition can be easily checked by "simulating" the operation of the maintenance station over the next time period, since both its time capacity and the residual maintenance times of the aircraft that will be grounded during the next time period are already known. This rule is also in direct alignment with the current practice of the HAF that strongly discourages the existence of unused maintenance time capacity. Even so, a case in which the service of a grounded aircraft does not begin in the next period even though it was anticipated to do so may actually come up. Such is the case, for example, when another aircraft that is grounded based on a subsequent decision begins its service first, due to a higher assigned priority.

When the grounding of an aircraft is decided, this aircraft flies its entire residual flight time in period t and enters the maintenance station for service at the beginning of period t+1; subsequently, the priority index of the squadron it belongs to is updated accordingly. After the complete set of aircraft that will be grounded at the beginning of period t+1 is determined, the flight times of the aircraft of each squadron that will be available at the beginning of period t+1 are obtained by solving a simple quadratic problem, as explained next.

Assume that the aircraft of squadron m that will be available in period t+1 (let their total number be A) are arranged on a flowchart in non-decreasing order of their residual flight times at the beginning of period t. Let F be the set that contains the indices of the aircraft of this squadron that are available in period t and will be grounded at the beginning of period t+1. These aircraft are not displayed on this graph, since they won't be available in period t+1. On the other hand, the aircraft of this squadron which are grounded during period t and will exit the maintenance station at the beginning of period t+1 are displayed with residual flight time equal to Y on this graph, but their maximum flight time in period t is set equal to 0, to ensure that no flight hours will be assigned to them. Let i be the index denoting the order of aircraft in this arrangement ($1 \leq i \leq A$), and let $s = Y/A$ be the slope of the diagonal. The target value for the residual flight time of the aircraft that appears in the i^{th} position at the beginning of period t+1 is equal to $i \cdot s$. Thus, the problem of deciding the flight time of each aircraft of squadron m reduces to the following quadratic programming problem:

$$\text{Min} \sum_{i=1}^{A} (y_{it+1} - is)^2$$

$$\text{s.t. } y_{it+1} = y_{it} - x_{it},\ i = 1,\dots, A$$

$$LS_{mt} \leq \sum_{i=1}^{A} x_{it} + \sum_{f \in F} y_{ft} \leq US_{mt}$$

$$x_{it} \leq X_{max},\ i = 1,\dots, A\text{: } i \text{ is available in period } t$$

$$x_{it} = 0,\ i = 1,\dots, A\text{: } i \text{ is grounded in period } t$$

$$y_{it+1} \geq Y_{min},\ i = 1,\dots, A$$

$$x_{it} \geq 0,\ i = 1,\dots, A$$

For simplicity, the above formulation denotes each aircraft using one index (i or f), instead of the indices m and n of the original formulation. The objective function minimizes the *total deviation index* of squadron m that will be realized at the beginning of period t+1, which is equal to the sum of squares of the deviations of the individual aircraft residual flight times from their corresponding target values. The first set of constraints updates the residual flight time of the aircraft at the beginning of period t+1. The next constraint ensures that the flight requirements of squadron m in period t will be satisfied (index f scans the aircraft of squadron m that will enter the station for service at the beginning of time period t+1). The next two sets of constraints impose upper bounds on the flight times of the aircraft, based on their status during the previous time period. The next set of constraints imposes a lower bound on the residual flight time of each available aircraft at the beginning of period t+1. Finally, the last set of constraints accounts for the non-negativity of the flight times. Note that the y_{it+1}'s and the x_{it}'s are decision variables in this formulation, whereas the y_{it}'s and the y_{ft}'s are known parameters. Setting $X_i = \min(X_{max}, y_{it} - Y_{min})$, for $i = 1,\dots, A$: i is available in period t, and $y_{it} = Y$, $X_i = 0$, for $i = 1,\dots, A$: i is grounded in period t, we obtain the following equivalent formulation:

$$\text{Min} \sum_{i=1}^{A} ((y_{it} - is) - x_{it})^2 \tag{4.3.1}$$

$$\text{s.t. } LS_{mt} - \sum_{f \in F} y_{ft} \leq \sum_{i=1}^{A} x_{it} \tag{4.3.2}$$

$$\sum_{i=1}^{A} x_{it} \leq US_{mt} - \sum_{f \in F} y_{ft} \tag{4.3.3}$$

$$0 \leq x_{it} \leq X_i,\ i = 1,\ldots, A \tag{4.3.4}$$

The problem defined by (4.3.1)-(4.3.4) is a quadratic program. The Hessian of the objective function is diagonal with all diagonal elements equal to 2; therefore, it is positive semi-definite, which implies that the objective function is convex. Hence, since all the constraints are linear, the KKT conditions (see Bazaraa et al., 2006) are necessary and sufficient for optimality. We give next a simple procedure called "Sweep" that can be utilized to obtain the optimal solution.

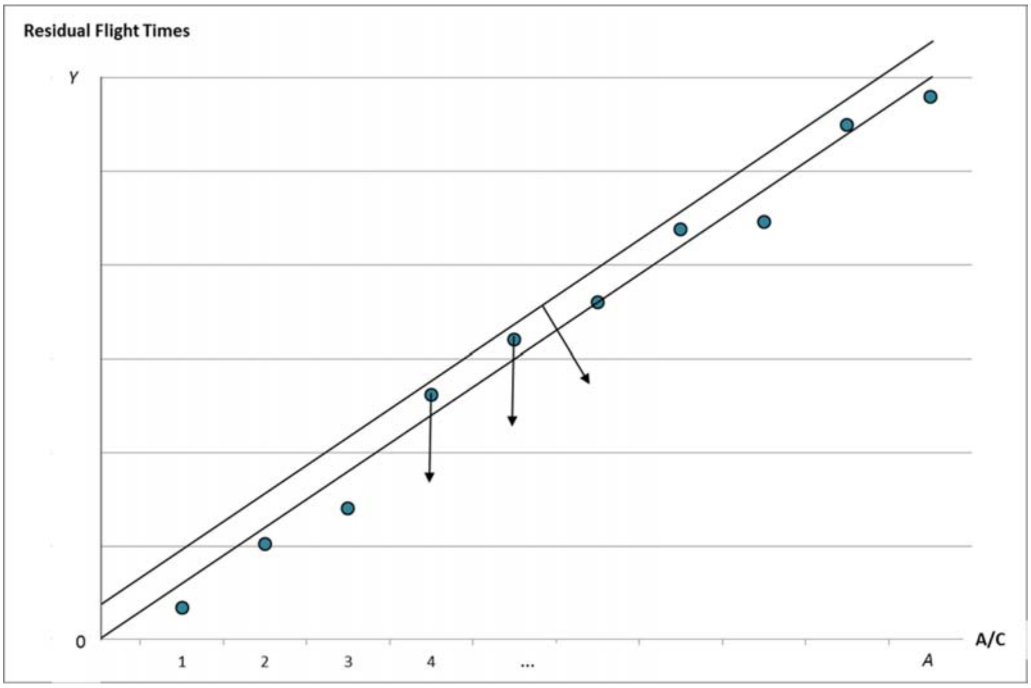

Figure 4.1: Illustration of Procedure "Sweep"

On the corresponding flowchart described above, consider a line parallel to the diagonal which is initially placed far enough to the top, so that all the aircraft lie below it, as shown in Figure 4.1 (in what follows, we do not distinguish between a point on the graph and the aircraft that this point represents). Assume now that this line starts moving towards the diagonal (and past it, while always remaining parallel to it), sweeping along *vertically* each aircraft that it comes across. Throughout this move, flight times are accordingly assigned to the aircraft in the order that they are swept by the line. If the flight time of an aircraft i reaches its maximum possible value, X_i, during this procedure, then the line should "disengage" this aircraft and continue its move without sweeping it further, to ensure that the resulting solution will remain feasible. Consider now the following 4 solutions that can be obtained during the application of this procedure:

1. The solution in which the sum of the assigned aircraft flight times is equal to $LS_{mt} - \sum_{f \in F} y_{ft}$. In what follows, we refer with "*LL*" to this sum.

2. The solution in which the sum of the assigned aircraft flight times is equal to $US_{mt} - \sum_{f \in F} y_{ft}$. In what follows, we refer with "*UL*" to this sum.

3. The solution in which each aircraft, i, is assigned its maximum possible flight time, X_i. In what follows, we refer with "*X*" to the sum of the assigned aircraft flight times of this solution.

4. The solution in which the sweeping line coincides with the diagonal. In what follows, we refer with "*D*" to the sum of the assigned aircraft flight times of this solution.

The following is a crucial and interesting result, utilized in the development of AFH:

Proposition 4.1. If the quantities *LL*, *UL*, *X* and *D* are arranged in non-decreasing order, then:

a) If, after taking into consideration any ties present, there does not exist an arrangement in which *LL* precedes *X*, then the problem defined by (4.3.1)-(4.3.4) is infeasible.

b) If an arrangement in which *LL* precedes *X* exists, then the optimal solution of the problem defined by (4.3.1)-(4.3.4) is the one obtained by Procedure Sweep when the sum of the assigned aircraft flight times becomes equal to the quantity that appears in the second place of this arrangement.

Proof. See Appendix A.

The application of Procedure Sweep produces the flight time of each available aircraft in each squadron of the unit. The same procedure is repeated successively for each time period, until the flight and maintenance plans for the entire planning horizon are produced. Based on the theory developed above, the detailed steps of AFH1 are introduced next using the additional notation presented below:

Sa_{mt} = number of available aircraft of squadron *m* in period *t*

Sx_{mt} = total flight time of squadron *m* in period *t*

Sy_{mt} = total residual flight time of squadron *m* at the beginning of period *t*

Sh_{mt} = total maintenance time of squadron *m* in period *t*

Sg_{mt} = total residual maintenance time of squadron *m* at the beginning of period *t*

ex_{mt} = number of aircraft of squadron *m* that exit the maintenance station at the end of period *t*

B_{res} = residual time capacity of the maintenance station

C_{res} = residual space capacity of the maintenance station

Aircraft Flowchart Heuristic 1 pseudocode

Step 0: Initialization

$C_{res} = C$;
for m = 1 to $|M|$ do
    $Sa_{m1} = 0$; $Sy_{m1} = 0$; $Sg_{m1} = 0$;
    for n = 1 to $|N_m|$ do
        $a_{mn1} = A1_{mn}$; $Sa_{m1} = Sa_{m1} + a_{mn1}$; $C_{res} = C_{res} - 1 + a_{mn1}$;
        $y_{mn1} = Y1_{mn}$; $Sy_{m1} = Sy_{m1} + y_{mn1}$;
        $g_{mn1} = G1_{mn}$; $Sg_{m1} = Sg_{m1} + g_{mn1}$;
    end for
    arrange in non-decreasing order of y_{mn1} the available aircraft of squadron m
    arrange in non-decreasing order of g_{mn1} the grounded aircraft of squadron m
end for
for t = 1 to T do
    Step 1: Production of maintenance plans for period *t*
    $B_{res} = B_t$;
    for m = 1 to $|M|$ do
        $ex_{mt} = 0$; $Sh_{mt} = 0$;
        for n = 1 to $|N_m|$ do
            $h_{mnt} = 0$;
        end for
    end for
    while $B_{res} > 0$ and at least one grounded aircraft waiting for service exists do
        if at least one grounded aircraft with interrupted service from previous periods exists
            $k = \underset{m \in M:\text{ squadron } m \text{ has at least one grounded aircraft with interrupted service}}{\arg\max} S_{mt}/(Sy_{mt} + (ex_{mt} * Y))$; # squadron with max priority index
            l = index of aircraft with lowest residual maintenance time among all grounded aircraft with interrupted service in squadron k
        else
            $k = \underset{m \in M:\text{ squadron } m \text{ has at least one grounded aircraft waiting for service}}{\arg\max} S_{mt}/(Sy_{mt} + (ex_{mt} * Y))$;
            l = index of aircraft with lowest residual maintenance time among all grounded aircraft waiting for service in squadron k
        end if
        if $B_{res} \geq g_{klt}$
        # if B_{res} suffices to finish the maintenance service of that aircraft
            $h_{klt} = g_{klt}$; $Sh_{kt} = Sh_{kt} + h_{klt}$; $B_{res} = B_{res} - h_{klt}$;
            $C_{res} = C_{res} + 1$; $ex_{mt} = ex_{mt} + 1$;
            # remove this aircraft from the set of grounded aircraft waiting for service
        else
        # if B_{res} does not suffice to finish the maintenance service of that aircraft
            $h_{klt} = B_{res}$; $Sh_{kt} = Sh_{kt} + h_{klt}$; $B_{res} = 0$;
        end if
    end while
    Step 2: Decision on aircraft that will be grounded in period t+1
    for m = 1 to $|M|$ do
        $Sx_{mt} = 0$;
        for n = 1 to $|N_m|$ do
            $x_{mnt} = 0$;
        end for
    end for
    set of candidate aircraft includes all available aircraft
    while $C_{res} > 0$ and the set of candidate aircraft is not empty do
        $k = \underset{m \in M:\text{ squadron } m \text{ contains at least one candidate aircraft}}{\arg\max} (S_{mt} - Sx_{mt})/(Sy_{mt} - Sx_{mt} + (ex_{mt} * Y))$; # squadron with max priority index
        l = index of candidate aircraft with lowest residual flight time in squadron k
        if the check for grounding this aircraft is successful
            $x_{klt} = y_{klt}$; $Sx_{kt} = Sx_{kt} + x_{klt}$; $C_{res} = C_{res} - 1$; # ground this aircraft

remove this aircraft from the set of candidate aircraft
else
remove all the available aircraft of squadron k from the set of candidate aircraft
end if
end while
Step 3: Production of flight plans for period t
for $m = 1$ to $|M|$ do
apply Procedure Sweep to compute the flight time (x_{mnt}) of each available aircraft
update Sx_{mt}
end do
Step 4: Update of status for period $t+1$
for $m = 1$ to $|M|$ do
$Sa_{mt+1} = Sa_{mt}$; $Sy_{mt+1} = Sy_{mt}$; $Sg_{mt+1} = Sg_{mt}$;
for $n = 1$ to $|N_m|$ do
if ($a_{mnt} == 0$ and $h_{mnt} == g_{mnt}$) # if this aircraft becomes available
$a_{mnt+1} = 1$; $Sa_{mt+1} = Sa_{mt+1} + 1$;
$y_{mnt+1} = Y$; $Sy_{mt+1} = Sy_{mt+1} + Y$;
$g_{mnt+1} = 0$; $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$;
else if ($a_{mnt} == 1$ and $x_{mnt} < y_{mnt}$) # if this aircraft retains availability
$a_{mnt+1} = 1$;
$y_{mnt+1} = y_{mnt} - x_{mnt}$; $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$;
$g_{mnt+1} = 0$;
else if ($a_{mnt} == 1$ and $x_{mnt} == y_{mnt}$) # if this aircraft is grounded
$a_{mnt+1} = 0$; $Sa_{mt+1} = Sa_{mt+1} – 1$;
$g_{mnt+1} = G$; $Sg_{mt+1} = Sg_{mt+1} + G$;
$y_{mnt+1} = 0$; $Sy_{mt+1} = Sy_{mt+1} - x_{mnt}$;
else if ($a_{mnt} == 0$ and $h_{mnt} < g_{mnt}$) # if this aircraft retains non-availability
$a_{mnt+1} = 0$;
$g_{mnt+1} = g_{mnt} - h_{mnt}$; $Sg_{mt+1} = Sg_{mt+1} - h_{mnt}$;
$y_{mnt+1} = 0$;
end if
end for
end for
end for

4.3.2 Aircraft Flowchart Heuristic 2 (AFH2)

The application of AFH2 is similar to that of AFH1, the only difference being that AFH2 does not utilize priority indices in order to arrange the aircraft of each squadron, but treats the aircraft as if they all belong to the same squadron. The next aircraft to receive maintenance service is always the one with the lowest residual maintenance time among all the grounded aircraft. Aircraft that are candidate for entering the maintenance station are considered in non-decreasing order of their residual flight times, independently of the squadron they belong to. The individual flight plan of each aircraft in each time period is produced by solving the quadratic problem defined by (4.3.1)-(4.3.4) once for each squadron. In the corresponding arrangement, however, the index i of each aircraft denotes its relative order when all the aircraft of the wing (not only those of the squadron this aircraft belongs to) are arranged in non-decreasing order of their residual flight times. Based on the theory developed above, the detailed steps of AFH2 are introduced next. Note that a rearrangement of the order of available aircraft takes place at the end of each time period in Step 4. This happens because the relative order between the available aircraft of different squadrons may change during the application of Procedure Sweep, even though it remains unchanged within each squadron.

Aircraft Flowchart Heuristic 2 pseudocode

Step 0: Initialization
Same with Step 0 in AFH1 with the additional requirement that the available aircraft of the wing should be arranged in non-decreasing order of y_{mn1} and the grounded aircraft of the wing should be arranged in non-decreasing order of g_{mn1} at the end of this step
for t = 1 to T do
    Step 1: Production of maintenance plans for period t
    $B_{res} = B_t$;
    for m = 1 to $|M|$ do
        $Sh_{mt} = 0$;
        for n = 1 to $|N_m|$ do
            $h_{mnt} = 0$;
        end for
    end for
    while $B_{res} > 0$ and at least one grounded aircraft waiting for service exists do
        if at least one grounded aircraft with interrupted service from previous periods exists
            l = index of aircraft with lowest residual maintenance time among all grounded aircraft with interrupted service
        else
            l = index of aircraft with lowest residual maintenance time among all grounded aircraft waiting for service
        end if
        k = index of the squadron the selected grounded aircraft belongs to
        if $B_{res} \geq g_{klt}$ # if B_{res} suffices to finish the maintenance service of the aircraft
            $h_{klt} = g_{klt}$; $Sh_{kt} = Sh_{kt} + h_{klt}$; $B_{res} = B_{res} - h_{klt}$;
            $C_{res} = C_{res} + 1$;
            remove this aircraft from the set of grounded aircraft waiting for service
        else
        # if B_{res} does not suffice to finish the maintenance service of this aircraft
            $h_{klt} = B_{res}$; $Sh_{kt} = Sh_{kt} + h_{klt}$; $B_{res} = 0$;
        end if
    end while
    Step 2: Decision on aircraft that will be grounded in period t+1
    for m = 1 to $|M|$ do
        $Sx_{mt} = 0$;
        for n = 1 to $|N_m|$ do
            $x_{mnt} = 0$;
        end for
    end for
    set of candidate aircraft includes all available aircraft
    while $C_{res} > 0$ and the set of candidate aircraft is not empty do
        l = index of candidate aircraft with lowest residual flight time
        k = index of the squadron that aircraft belongs to
        if the grounding check for this aircraft is successful
            $x_{klt} = y_{klt}$; $Sx_{kt} = Sx_{kt} + x_{klt}$; $C_{res} = C_{res} - 1$;
            remove this aircraft from the set of candidate aircraft
        else
            remove all the available aircraft of squadron k from the set of candidate aircraft
        end if
    end while
    Step 3: Production of flight plans for period t
    Same with Step 3 in AFH1
    Step 4: Update for period t+1

Same with Step 4 in AFH1 with the additional requirement that the available aircraft of the wing should be arranged in non-decreasing order of y_{mnt+1} after the status update of the aircraft of each squadron

end for

4.3.3 Horizon Splitting Heuristic (HSH)

The second heuristic procedure that we propose for the solution of large FMP instances utilizes the simple idea of splitting the original planning horizon into several consecutive ones, and applying an FMP optimization model to each of them. The ending state of the system in each sub-horizon becomes the beginning state of the next one, and so on. The smaller horizons do not necessarily need to have equal lengths. The quality of the solution obtained this way is expected to be inferior to the one obtained when the problem is solved up front for all the periods of the original planning horizon. On the other hand, the total computational time needed in order to reach a solution is expected, in general, to decrease, especially as the length of the smaller horizons decreases. This is mainly because the computational effort needed to reach an optimal solution is expected (in general, but not necessarily always) to increase as the size of the problem increases.

4.4 Computational implementation

In this section, we analyze the worst-case computational complexity of AFH1 and AFH2, and we present computational results evaluating the performance of AFH1, AFH2 and HSH on randomly generated FMP instances. Of course, each subproblem solved by HSH is itself an FMP optimization model, and therefore, exhibits the same complexity.

4.4.1 Computational complexity of AFH1 and AFH2

In order to analyze the computational complexity analysis of AFH1, we prove an interesting result first.

Lemma 4.1. The problem defined by (4.3.1)-(4.3.4) can be solved in time O(A), where A is the total number of variables x_{it}.

Proof. See Appendix B.

Let $N_{\max} = \max_{m \in M} |N_m|$ and $N = \sum_{m \in M} N_m$. Propositions 4.2 and 4.3 utilize Lemma 4.1 in order to analyze the computational complexity of AFH1 and AFH2, respectively.

Proposition 4.2. AFH1 requires time $O(\max(|M|N_{max}\log N_{max}, T|M|N_{max}, T|M|^2, T|M|C^2, TC^3))$.

Proof. The dominating operation in Step 0 is the arrangement of the available and the grounded aircraft that requires time $O(N_{max}\log N_{max})$ and is executed $|M|$ times. Therefore, the total time that Step 0 requires is $O(|M|N_{max}\log N_{max})$. The initialization commands in the first 7 lines of Step 1 require time $O(|M|N_{max})$. The check in the while command of Step 1 requires time O(1). Checking if an aircraft with incomplete service from previous periods exists requires time O(1). Finding the squadron with the highest priority index among all the squadrons with aircraft with interrupted service (or all the squadrons with aircraft waiting for service) requires time $O(\min(i,|M|))$, where i is the number of grounded aircraft ($1 \le i \le C$). Selecting the appropriate aircraft from that squadron requires time O(1), since the grounded aircraft of each squadron are always sorted in non-decreasing order of their residual maintenance times. The if-else clause that decides the maintenance time of the selected aircraft requires time O(1). Since $\min(i,|M|) = C$ in the worst case, the while-loop of Step 1 is repeated at most C times, with each repetition i ($1 \le i \le C$) requiring time $O(C-i+1)$ for selecting the appropriate squadron and aircraft, and time O(1) for deciding its maintenance time. Since $C + (C-1) +...+ 1 = \frac{C(C+1)}{2}$, the total time required for this is $O(C^2)$. Note that this bound cannot be improved by sorting in advance the squadrons with respect to their priority index, since one of these indices always changes in each iteration, imposing a rearrangement. Therefore, Step 1 requires time $O(|M|N_{max}) + O(C^2) = O(\max(|M|N_{max}, C^2))$ in total.

The initialization commands in the first 7 lines of Step 2 require time $O(|M|N_{max})$. For the while-loop that follows, we distinguish two cases. The first case is when this loop is repeated $|M|+C-1$ times ($C-1$ successful checks for grounding an aircraft first, followed either by $|M|-1$ unsuccessful and one successful, or by $|M|$ unsuccessful). This is the maximum possible number of times that this loop can be repeated. The sequence of checks implies that the maintenance station is initially empty. Moreover, simulating the operation of the maintenance station at the presence of i grounded aircraft requires time $O(i^2)$, since there is a total of at most i iterations involved, where iteration k ($1 \le k \le i$) selects out of at most $i-k+1$ squadrons the one with the highest priority index (again a prearrangement is not sufficient to improve the bound, since one of these indices changes in each iteration) and assigns a maintenance time to a grounded aircraft that belongs to that squadron. It follows that repetition i ($1 \le i \le C-1$) requires time $O(|M|)$ for selecting the squadron with the highest priority index and time $O(i^2)$ for checking the condition for grounding the associated aircraft, while each repetition i of the next $|M|$ ones ($C \le i \le |M|+C-1$) requires time $O(|M|+C-i)$ for selecting the squadron with the highest priority index and time $O((C-1)^2)$ for checking the condition for grounding the associated aircraft. Therefore, the total time required for the while-loop in this case is $O((|M|+1^2) + (|M|+2^2) +...+ (|M|+(C-1)^2) + (|M|+(C-1)^2) + (|M|-1+(C-1)^2) +...+ (1+(C-1)^2)) = O((C-1)|M| + (1^2+2^2+...+(C-1)^2) + (1+2+...+|M|) +$

$|M|(C\text{-}1)^2) = O(C|M| - |M| + \frac{(C-1)C(2(C-1)+1)}{6} + \frac{(|M|)(|M|+1)}{2} + |M|C^2 + |M| -2C|M|) = O(|M|^2+|M|C^2+C^3) = O(\max(|M|^2,|M|C^2,C^3))$. The second case is when the while-loop is repeated $|M|$ times ($|M|$ unsuccessful checks for grounding an aircraft from $|M|$ distinct squadrons). In this case, repetition i ($1 \leq i \leq |M|$) requires time $O(|M|\text{-}i+1)$ for selecting the squadron with the highest priority index, leading to a total bound of $O(|M| + |M|\text{-}1 + ... + 1) = O(\frac{|M|(|M|+1)}{2}) = O(|M|^2)$ for all $|M|$ selections (since the selected squadron is dropped at the end of each iteration, this bound can be improved to $O(|M|\log|M|)$ by sorting in advance the $|M|$ squadrons with respect to their priority index, but this is neither known in advance, nor can it improve the overall bound of Step 2 computed next). Additionally, since one aircraft is considered for entering the maintenance station at each iteration, there are at most C-1 grounded aircraft; thus, the time required for this check is $O((C\text{-}1)^2)$. Hence, the total time required for the while-loop is $O(|M|^2 + |M|(C\text{-}1)^2) = O(\max(|M|^2,|M|C^2))$ in this case. Thus, considering the worst of the above two cases, Step 2 requires time $O(\max(|M|^2,|M|C^2,C^3,|M|^2,|M|C^2) = O(\max(|M|^2,|M|C^2,C^3))$ in total. In Step 3, solving $|M|$ times the problem defined by (4.3.1)-(4.3.4) requires time $O(|M|N_{max})$ in total. Finally, updating the status of the system for period t+1 in Step 4 requires time $O(|M|N_{max})$. Since Step 0 is executed once and each of Steps 1-4 is executed T times, the total time required by AFH1 is $O(|M|N_{max}\log N_{max} + T(|M|N_{max} + C^2 + |M|^2 + |M|C^2 + C^3 + |M|N_{max} + |M|N_{max}) = O(\max(|M|N_{max}\log N_{max},T|M|N_{max},T|M|^2,T|M|C^2,TC^3))$.

Proposition 4.3. AFH2 requires time $O(\max(|M|N_{max}\log N_{max},TC,T|M|N_{max}\log|M|)$.

Proof. The arrangement of the available and the grounded aircraft of each squadron in Step 0 requires time $O(N_{max}\log N_{max})$ and is executed $|M|$ times in total. The arrangement of the available and the grounded aircraft of the wing requires time $O(N\log|M|)$, since the aircraft of each squadron are already sorted. These are the two dominating operations in Step 0; therefore, the total time that Step 0 requires is $O(|M|N_{max}\log N_{max}) + O(N\log|M|) = O(|M|N_{max}\max(\log N_{max}, \log|M|))$. The initialization commands in the first 7 lines of Step 1 require time $O(|M|N_{max})$. The check in the while command of Step 1 requires time O(1). Checking if an aircraft with interrupted service from previous periods exists requires time O(1). Selecting the grounded aircraft that should receive maintenance service next requires time O(1), since the grounded aircraft of the wing are always sorted in non-decreasing order of their residual maintenance times. The if-else clause that decides the maintenance time of the selected aircraft requires time O(1). Since the while-loop of Step 1 is repeated C times in the worst-case, Step 1 requires time $O(|M|N_{max}) + O(C) = O(\max(|M|N_{max},C))$ in total.

The initialization commands in the first 7 lines of Step 2 require time $O(|M|N_{max})$. For the while-loop that follows, we distinguish two cases. The first case is when this loop is repeated $|M|+C$-1 times (C-1 successful checks for grounding an aircraft first, followed either by $|M|$-1 unsuccessful and one successful, or by $|M|$ successful). This is the maximum possible number of times that this loop can be repeated. The sequence of checks implies that the maintenance station is initially empty. In this case, repetition i ($1 \leq i \leq |M|+C-1$) requires time O(1) for selecting the appropriate aircraft and time O(1) for checking whether it should be grounded. Therefore, the total time required for the while-loop is $O(|M|+C)$. The second case is when the while-loop is repeated $|M|$ times ($|M|$ unsuccessful checks for grounding an aircraft from $|M|$ distinct squadrons). In this case, each repetition requires time O(1) for selecting the appropriate aircraft, and time O(1) for checking whether it should be grounded (since the aircraft are treated as if they all belong to the same squadron, the "simulation" of the operation of the maintenance station is trivial). Hence, the total time required for the while-loop is $O(|M|)$. Thus, considering the worst of the above two cases, Step 2 requires time $O(|M|+C)$ in total. In Step 3, solving $|M|$ times the problem defined by (4.3.1)-(4.3.4), requires time $O(|M|N_{max})$ in total. In Step 4, updating the status of the system for period t+1 requires time $O(|M|N_{max})$ and sorting the available aircraft of the wing requires time $O(N\log|M|)$, since the available aircraft of each squadron are already sorted. Therefore, the total time required by Step 4 is $O(|M|N_{max}+ N\log|M|) = O(|M|N_{max}\log|M|)$. Since Step 0 is executed once and each of Steps 1-4 is repeated T times, the total time required by AFH2 is $O(|M|N_{max}\log N_{max} + |M|N_{max}\log|M| + T(\max(|M|N_{max},C) + (|M|+C) + |M|N_{max} + |M|N_{max}\log|M|)) = O(\max(|M|N_{max}\log N_{max},TC,T|M|N_{max}\log|M|)$.

Note that, in most of the cases, the available aircraft of the wing will almost be sorted at the end of Step 4. Therefore, sorting them will require time $O(A)$ (using a sorting technique such as insertion-sort); in such a case, the total time required by AFH2 will be $O(\max(|M|N_{max}\log N_{max},|M|N_{max}\log|M|, T|M|N_{max}, TC))$.

4.4.2 Computational results

We implemented AFH1 and AFH2 in C/C++ and tested their performance against that of HSH and the FMP model of Section 4.2. Since their design does not depend on the adopted objective function, we only applied AFH1 and AFH2 once on each random problem instance. For the solution of FMP and HSH, we utilized version 10.1 of AMPL/CPLEX (see Fourer et al., 2002), with default values where possible. We performed all the experiments on a Dual Xeon server with a 2 GHz processor and 2 GB system memory.

We used 12 different combinations for the values of $|M|$, $|N_m|$ and T and solved 10 random problem instances for each of them. We chose a smaller size for the problem instances of the first 8 combinations, in order to enable their exact solution with the FMP model, and evaluate this way the quality of the solutions produced by the heuristics. On the other hand, the remaining 4 combinations correspond to actual wing sizes encountered in the HAF, for which the FMP model cannot find the optimal solution in reasonable times. Since the data pertaining to real-world problem instances are strictly confidential, we mainly use these problem instances to infer the performance of the heuristics on real problem cases.

The value of $|N_m|$ was always the same across all squadrons. For $|M|$, we used the values 2 and 3, since typical HAF wings consist of 2 or 3 squadrons. In the first 8 combinations, the value of $|N_m|$ was taken equal to 6-10 when $|M|$ = 2, and 6-8 when $|M|$ = 3. In the remaining 4 combinations, we considered the values 20 and 25 for $|N_m|$, to reflect the fact that a typical HAF squadron may contain up to 20-25 aircraft. The value of T was always taken equal to 6, since the flight requirements are typically issued for a planning horizon of 6 monthly periods.

The required flight time for each squadron and period combination was a random number distributed uniformly in the interval $[16|N_m|, 21|N_m|]$. The time capacity of the maintenance station in each time period was a random number distributed uniformly in the interval $[21|M||N_m|, 26|M||N_m|]$, and the space capacity was set equal to $0.1|M||N_m|$, rounded up to the nearest integer. These figures correspond to actual FMP configurations encountered in the HAF. We generated the number of grounded aircraft randomly, using a discrete probability function that considered integer values between 0 and C, inclusive. This distribution associated higher values with higher probabilities, in order to favor more challenging problems. We set parameters Y and G equal to their actual values, i.e., 300 and 320 hours, respectively. The residual flight time of each available aircraft was a random number distributed uniformly in the interval $[Y_{min},Y]$, whereas the residual maintenance time of each grounded aircraft was a random number distributed uniformly in the interval $[G_{min},G]$. We used actual values drawn from the real application for the remaining problem parameters, i.e., L = 0.9, U = 1.1, X_{max} = 50, $Y_{\min}$ = 0.1 and G_{min} = 0.1. We performed several checks to ensure that each randomly generated problem instance was feasible.

First, we applied FMP and HSH as single objective models, adopting each of the 6 objective functions presented in Section 4.2. The ideal value (see Ehrgott, 2005) of the corresponding objective was obtained this way. Although the ideal values are useful for evaluating the quality of the solutions produced by AFH1, AFH2 and HSH, a feasible solution that simultaneously attains them will rarely exist. To assess this quality, we also solved each of the random problem instances using the weighted sums approach (see Steuer, 1986) with two uniform weight combinations. In the first case, we introduced positive weights $w_1 = w_2 = w_3 = w_4 = 0.25$ and we solved the FMP model with objective $Z_A = 0.25(Y/2)z_1 + 0.25|M|(Y/2)z_2 + 0.25z_3 + 0.25|M|z_4$. We introduced the constants $(Y/2)$ and $|M|$ for scaling reasons, since the residual flight time of an available aircraft is equal to $Y/2$ on the average, and increasing the fleet availability of every squadron by 1 is equivalent to increasing the fleet availability of the wing by $|M|$. In the second case, we introduced positive weights $w_5 = w_6 = 0.5$ and we solved the FMP model with objective $Z_B = 0.5(Y/2)z_5 + 0.5z_6$. We also tested the performance of HSH using the objectives Z_A and Z_B, by splitting the original 6-period time horizon into two 3-period ones and applying the FMP model twice.

Tables 4.1-4.3 compare the six criteria values provided by AFH1 with their corresponding ideal values and show the fleet availabilities that the obtained solutions provide. More specifically, out of the four columns that pertain to the same criterion, the first two show the average and maximum percentage difference between each criterion value and its corresponding ideal value, whereas the next two show the average and minimum associated availability. Results for the differences from the ideal values are not reported for the last 4 sets of problem instances, since it was not possible to apply the FMP model in order to obtain the ideal value of each associated criterion for them. The availabilities are not expressed in absolute values, but as a percentage over their theoretically maximum possible value. For example, the theoretically maximum possible value of objective z_1 is equal to the total number of aircraft, $|M||N_m|$.

Table 4.1: Quality and fleet availability of the solutions provided by AFH1 for objectives z_1 and z_2

$\|M\|$	$\|N_m\|$	z_1 % from Ideal		z_1 % availability		z_2 % from Ideal		z_2 % availability	
		Avg	**Max**	**Avg**	**Min**	**Avg**	**Max**	**Avg**	**Min**
2	**6**	13.64	16.67	83.33	83.33	23.00	33.33	71.67	66.67
2	**7**	10.62	14.29	85.71	85.71	18.25	28.57	76.19	71.43
2	**8**	10.17	12.50	87.50	87.50	18.21	25.00	77.50	75.00
2	**9**	10.59	11.11	88.89	88.89	18.89	22.22	80.00	77.78
2	**10**	13.39	15.00	86.11	85.00	17.90	30.00	81.11	70.00
3	**6**	11.11	11.11	88.89	88.89	31.67	33.33	68.33	66.67
3	**7**	12.95	14.29	86.19	85.71	23.10	28.57	74.29	71.43
3	**8**	12.50	12.50	87.50	87.50	26.39	37.50	73.61	62.50
2	**20**			87.75	87.50			82.50	80.00
2	**25**			88.40	88.00			82.80	80.00
3	**20**			88.50	88.33			79.50	75.00
3	**25**			89.33	89.33			79.20	76.00

Table 4.2: Quality and fleet availability of the solutions provided by AFH1 for objectives z_3 and z_4

$\|M\|$	$\|N_m\|$	z_3 % from Ideal		z_3 % availability		z_4 % from Ideal		z_4 % availability	
		Avg	**Max**	**Avg**	**Max**	**Avg**	**Min**	**Avg**	**Min**
2	**6**	3.97	9.15	3.59	9.15	35.94	30.21	41.83	36.95
2	**7**	5.20	28.20	4.16	19.46	34.78	26.46	41.77	30.85
2	**8**	1.83	10.23	0.16	1.62	39.57	27.09	47.27	40.37
2	**9**	4.49	10.19	4.12	11.09	42.80	33.14	48.87	42.37
2	**10**	3.35	11.38	2.56	7.81	42.09	33.35	47.19	42.64
3	**6**	2.98	12.26	2.25	13.81	34.51	29.07	46.03	40.36
3	**7**	2.99	15.41	3.11	14.77	37.16	30.40	44.75	36.80
3	**8**	1.34	4.59	0.53	4.63	36.63	26.54	44.47	38.49
2	**20**					42.04	34.74	45.66	40.97
2	**25**					41.23	34.47	44.98	40.09
3	**20**					38.10	28.41	43.74	39.19
3	**25**					38.96	32.98	44.15	41.35

Table 4.3: Quality and fleet availability of the solutions provided by AFH1 for objectives z_5 and z_6

$\lvert M\rvert$	$\lvert N_m\rvert$	z_5 % from Ideal		z_5 % availability		z_6 % from Ideal		z_6 % availability	
		Avg	Max	Avg	Min	Avg	Max	Avg	Min
2	**6**	12.15	12.68	87.36	86.11	5.18	9.52	46.71	42.55
2	**7**	10.94	13.10	88.23	85.71	4.00	20.37	46.04	34.32
2	**8**	9.52	11.46	90.10	88.54	3.23	6.85	49.68	44.68
2	**9**	8.99	10.19	90.93	89.81	4.50	9.03	50.76	44.35
2	**10**	10.66	13.33	89.26	86.67	4.40	8.00	50.18	45.46
3	**6**	9.17	10.19	90.83	89.81	2.15	8.77	48.77	43.22
3	**7**	10.41	12.70	89.44	87.30	3.52	13.68	48.26	40.22
3	**8**	10.26	11.11	89.74	88.89	2.75	3.83	47.33	42.93
2	**20**			90.46	89.58			47.75	43.28
2	**25**			90.83	89.33			47.30	43.13
3	**20**			90.78	90.28			46.64	43.20
3	**25**			90.89	90.44			46.91	44.04

Tables 4.4-4.9 present similar results for AFH2 and HSH. Results for HSH are not reported at all for the last 4 size combinations, since the application of HSH on these problems is impractical due to its excessive computational effort. The extra columns labeled "IF" in Tables 4.7-4.9 show the number of instances in the associated combination for which HSH did not return a feasible solution. The computation of the results reported in these tables was based only on the instances for which HSH reached a feasible solution.

Table 4.4: Quality and fleet availability of the solutions provided by AFH2 for objectives z_1 and z_2

$\lvert M\rvert$	$\lvert N_m\rvert$	z_1 % from Ideal		z_1 % availability		z_2 % from Ideal		z_2 % availability	
		Avg	Max	Avg	Min	Avg	Max	Avg	Min
2	**6**	13.64	16.67	83.33	83.33	26.33	33.33	68.33	66.67
2	**7**	10.62	14.29	85.71	85.71	19.84	28.57	74.60	71.43
2	**8**	10.17	12.50	87.50	87.50	19.46	25.00	76.25	75.00
2	**9**	10.59	11.11	88.89	88.89	20.14	22.22	78.89	77.78
2	**10**	13.11	15.00	86.00	85.00	22.33	30.00	76.00	70.00
3	**6**	11.11	11.11	88.89	88.89	28.33	33.33	71.67	66.67
3	**7**	12.95	14.29	86.19	85.71	24.76	28.57	72.86	71.43
3	**8**	12.12	12.50	87.50	87.50	27.68	37.50	71.25	62.50
2	**20**			89.25	87.50			83.00	80.00
2	**25**			90.00	90.00			84.80	84.00
3	**20**			90.17	90.00			81.50	80.00
3	**25**			90.53	89.33			84.00	80.00

Table 4.5: Quality and fleet availability of the solutions provided by AFH2 for objectives z_3 and z_4

$\|M\|$	$\|N_m\|$	z_3 % from Ideal		z_3 % availability		z_4 % from Ideal		z_4 % availability	
		Avg	Max	Avg	Max	Avg	Max	Avg	Min
2	**6**	0.97	4.56	7.34	23.35	7.34	23.35	68.33	66.67
2	**7**	6.66	21.02	15.12	38.01	15.12	38.01	74.60	71.43
2	**8**	1.57	5.02	11.31	30.52	11.31	30.52	76.25	75.00
2	**9**	7.24	18.58	14.84	26.26	14.84	26.26	78.89	77.78
2	**10**	4.51	11.18	10.15	21.91	10.15	21.91	76.00	70.00
3	**6**	2.51	11.08	14.50	35.09	14.50	35.09	71.67	66.67
3	**7**	3.10	20.33	16.57	31.76	16.57	31.76	72.86	71.43
3	**8**	1.65	5.51	16.01	38.93	16.01	38.93	71.25	62.50
2	**20**							83.00	80.00
2	**25**							84.80	84.00
3	**20**							81.50	80.00
3	**25**							84.00	80.00

Table 4.6: Quality and fleet availability of the solutions provided by AFH2 for objectives z_5 and z_6

$\|M\|$	$\|N_m\|$	z_5 % from Ideal		z_5 % availability		z_6 % from Ideal		z_6 % availability	
		Avg	Max	Avg	Min	Avg	Max	Avg	Min
2	**6**	13.83	14.08	85.69	84.72	3.92	7.59	47.32	43.93
2	**7**	10.00	13.10	89.02	86.90	6.15	18.06	47.31	38.19
2	**8**	9.94	11.46	89.69	88.54	4.25	7.31	49.13	45.55
2	**9**	8.99	10.19	90.93	89.81	6.67	13.29	49.60	44.16
2	**10**	9.85	10.83	90.00	89.17	6.27	11.42	49.44	45.56
3	**6**	9.44	10.19	90.56	89.81	2.80	7.61	48.44	42.83
3	**7**	9.38	11.11	90.48	88.89	4.59	15.06	47.73	40.22
3	**8**	9.80	10.42	90.14	89.58	3.41	7.96	46.63	44.06
2	**20**			91.67	91.25			47.59	43.82
2	**25**			92.13	91.67			47.22	43.33
3	**20**			92.36	92.22			46.33	43.34
3	**25**			92.60	92.44			46.75	44.04

Table 4.7: Quality and fleet availability of the solutions provided by HSH for objectives z_1 and z_2

$\|M\|$	$\|N_m\|$	z_1 % from Ideal			z_1 % availability		z_2 % from Ideal			z_2 % availability	
		Avg	Max	IF	Avg	Min	Avg	Max	IF	Avg	Min
2	**6**	0.00	0.00	2	93.75	83.33	0.00	0.00	0	93.75	83.33
2	**7**	0.00	0.00	2	92.86	92.86	1.79	14.29	2	85.71	85.71
2	**8**	0.00	0.00	3	92.86	87.50	0.00	0.00	2	87.50	87.50
2	**9**	0.00	0.00	0	94.44	94.44	0.00	0.00	0	88.89	88.89
2	**10**	0.00	0.00	0	93.75	90.00	0.00	0.00	0	90.00	90.00
3	**6**	0.00	0.00	1	94.44	94.44	0.00	0.00	0	83.33	83.33
3	**7**	0.00	0.00	0	92.86	90.48	0.00	0.00	0	85.71	85.71
3	**8**	0.00	0.00	0	94.44	91.67	0.00	0.00	0	87.50	87.50

Table 4.8: Quality and fleet availability of the solutions provided by HSH for objectives z_3 and z_4

$\|M\|$	$\|N_m\|$	z_3 % from Ideal			z_3 % availability		z_4 % from Ideal			z_4 % availability	
		Avg	Max	Avg	Max	Avg	Max	Avg	IF	Avg	Min
2	**6**	12.38	20.63	16.41	33.16	16.41	33.16	16.41	0	93.75	83.33
2	**7**	7.72	14.98	4.52	14.25	4.52	14.25	4.52	2	85.71	85.71
2	**8**	5.97	9.58	12.73	30.93	12.73	30.93	12.73	2	87.50	87.50
2	**9**	9.36	17.70	7.63	17.82	7.63	17.82	7.63	0	88.89	88.89
2	**10**	4.61	12.29	6.52	21.71	6.52	21.71	6.52	0	90.00	90.00
3	**6**	7.67	22.64	5.93	21.93	5.93	21.93	5.93	0	83.33	83.33
3	**7**	3.01	6.97	11.16	36.72	11.16	36.72	11.16	0	85.71	85.71
3	**8**	3.27	10.08	7.25	17.36	7.25	17.36	7.25	0	87.50	87.50

Table 4.9: Quality and fleet availability of the solutions provided by HSH for objectives z_5 and z_6

$\|M\|$	$\|N_m\|$	z_5 % from Ideal			z_5 % availability		z_6 % from Ideal			z_6 % availability	
		Avg	Max	IF	Avg	Min	Avg	Max	IF	Avg	Min
2	**6**	0.18	1.41	2	99.31	97.22	10.89	14.96	1	36.98	32.54
2	**7**	0.15	1.20	2	96.73	96.43	6.60	10.93	3	44.24	29.56
2	**8**	0.00	0.00	3	95.68	94.79	6.05	11.43	1	47.04	40.90
2	**9**	0.00	0.00	0	96.67	96.30	7.10	11.90	0	49.52	39.72
2	**10**	0.00	0.00	0	96.46	95.00	5.59	10.52	6	46.91	40.32
3	**6**	0.00	0.00	1	96.30	96.30	6.70	12.28	1	44.30	37.67
3	**7**	0.00	0.00	0	96.03	95.24	5.55	9.21	1	50.13	37.58
3	**8**	0.00	0.00	0	95.83	95.83	5.18	10.32	1	45.13	36.81

Several interesting observations can be made based on the results of the above tables. The results of Tables 4.1-4.3 show that the quality of the solutions produced by AFH1 is quite satisfactory for criterion z_1 (approximately 12% difference from the corresponding ideal values on average), less satisfactory for criterion z_2 (approximately 22% difference from the corresponding ideal values on average), very satisfactory for criteria z_3 and z_4 (approximately 2-3% difference from the corresponding ideal values on average), quite satisfactory for criterion z_5 (approximately 10% difference from the corresponding ideal values on average), and very satisfactory for criterion z_6 (approximately 4% difference from the corresponding ideal values on average). The solution quality of some particular instances is significantly worse than average, as denoted by the large difference between columns “Avg” and “Max”. The results of Tables 4.4-4.6 show that the behavior of AFH2 is similar, the only difference being that the solution quality for criterion z_4 is more comparable to that of criterion z_1 (approximately 12% difference from the corresponding ideal values on average).

The availabilities that we report for criteria z_3, z_4 and z_6 are significantly lower than those reported for criteria z_1, z_2 and z_5. This is not surprising, since the criteria z_3, z_4 and z_6 pertain to the residual flight time availability which cannot be improved dramatically with respect to the value that it has at the beginning of the planning horizon. This is due to the presence of the flight load requirements, and the fact that the replenishment of the flight hours through the completion of maintenance service is constrained by the limited time capacity of the maintenance station. To provide further intuition on this, note that any aircraft which is available at the initial state of the system is expected to have residual flight time availability equal to $(Y+Y_{min})/2$ = 150.05 hours on the average. Coupled with the fact that some of the unit’s aircraft are initially expected to be grounded, this implies that the initial residual flight time availability of the unit is expected to be smaller than 50% of its maximum possible value. On the other hand, starting with a large number of available aircraft, it is easier to retain a high level of aircraft availability for the entire planning horizon, leading to drastically higher values for criteria z_1, z_2 and z_5.

As shown in Tables 4.7-4.9, HSH exhibits a rather myopic behavior and is not always able to return a feasible solution, mainly because it treats each sub-horizon separately. This is due to the fact that the decisions made in an early sub-horizon may turn out to be too restricting, leaving the system in an inadequate state that is incapable of satisfying the flight requirements of the following sub-horizon. This issue can be resolved by a more conservative planning over the initial periods of the planning horizon. In the actual combat wing that we studied, $|N_m|$ and X_{max} were always large enough, making it always possible to satisfy this flight load using only a subset of the available aircraft. Thus, finding a feasible problem solution was hardly troublesome. In general, however, a careful design should address accordingly such difficulties that may arise.

Aside from this issue, it is impressive that HSH was able to find the ideal value of criteria z_1 and z_2 in almost all the instances for which it reached a feasible solution. On the other hand, the quality of the solutions produced by HSH is very satisfactory for criterion z_3 (approximately 6-7% difference from the corresponding ideal values on average), and a little less satisfactory for z_4 (approximately 9-10% difference from the corresponding ideal values on average). HSH was not able to find a feasible solution for criterion z_5 in 8 out of 80 instances, but found the ideal value of the associated criterion in almost all the instances for which it reached a feasible solution. Additionally, the quality of the solutions produced by HSH is very satisfactory for criterion z_6 (approximately 6-7% difference from the corresponding ideal values on average).

In Tables 4.10 and 4.11, we perform a comparison of the optimal values of the weighted sum objectives Z_A and Z_B provided by AFH1, AFH2, HSH and FMP. In order to tie these results with those of Tables 4.1-4.9, we also compare in the last two columns of these tables the optimal Z_A and Z_B values provided by FMP, with the ones that result when the ideal values of the corresponding criteria are used. Results for the last 4 sets of problem instances are not reported, since the large size of these problem instances renders the application of FMP and HSH impractical.

Table 4.10: Comparison of optimal Z_A values

$\|M\|$	$\|N_m\|$	AFH1 % from FMP		AFH2 % from FMP		HSH % from FMP			FMP % from Ideal	
		Avg	Max	Avg	Max	Avg	Max	IF	Avg	Max
2	**6**	5.30	9.18	6.30	10.44	14.61	16.86	2	6.73	11.24
2	**7**	5.97	14.86	9.09	14.01	7.58	19.25	2	4.14	7.88
2	**8**	4.83	6.16	7.64	12.87	3.30	5.81	3	3.28	5.08
2	**9**	6.03	9.78	9.60	11.71	5.12	6.46	5	4.01	4.91
2	**10**	5.14	8.22	8.71	13.03	5.10	10.52	2	4.45	7.21
3	**6**	5.92	12.75	7.46	13.42	3.09	4.73	4	7.49	12.36
3	**7**	6.37	8.39	9.88	14.40	5.63	7.98	6	5.52	7.12
3	**8**	5.57	8.90	9.70	18.03	4.39	6.74	7	5.46	6.96

Table 4.11: Comparison of optimal Z_B values

$\lvert M\rvert$	$\lvert N_m\rvert$	AFH1 % from FMP		AFH2 % from FMP		HSH % from FMP			FMP % from Ideal	
		Avg	Max	Avg	Max	Avg	Max	IF	Avg	Max
2	**6**	3.45	5.34	3.67	5.63	7.31	9.17	2	5.47	5.76
2	**7**	4.71	14.71	5.33	13.97	2.82	6.55	2	3.14	4.15
2	**8**	4.11	5.82	4.87	6.98	2.56	6.09	3	2.37	3.59
2	**9**	3.60	5.52	4.76	8.64	2.40	3.66	5	3.29	4.42
2	**10**	3.86	5.46	4.46	6.25	3.68	4.53	2	3.78	4.33
3	**6**	2.38	6.06	2.85	5.88	3.08	4.14	4	3.45	4.64
3	**7**	3.54	7.90	3.55	7.66	4.07	5.71	6	3.78	4.61
3	**8**	2.74	3.59	2.83	5.37	4.69	6.51	7	4.00	4.58

Applying FMP with a weighted sums objective produces a high-quality solution, which is additionally guaranteed to be non-dominated (see Steuer, 1986) for the associated multi-objective problem. Moreover, the Z_A values of the optimal solutions produced by FMP appear to be close (approximately 3-5% on average) to the corresponding Z_A values that result when the ideal value of each criterion is used. This is because the objectives of the model are not in direct conflict with each other, but there exists a certain degree of synergy among them. It should be noted however, that if we replace one of the objectives with a corresponding lower bound constraint, the model will only focus on ensuring this bound for that objective, without any special concern for optimizing it.

The weighted sums objective function value of the solutions produced by AFH1 appears to be close (approximately 5% on average for Z_A and 3.5% on average for Z_B) to the corresponding Z_A and Z_B values provided by FMP. On the other hand, AFH2 seems to perform slightly worse than AFH1 with an average difference of approximately 8.5% for Z_A and 4% for Z_B. These percentages do not remain constant, but exhibit significant variance. Despite the existence of this variance, AFH1 and AFH2 perform quite satisfactorily on average. HSH seems to perform comparably to AFH1 and AFH2, with an average difference of approximately 6% for Z_A and 4% for Z_B, and variance of similar magnitude. The difficulty in reaching feasibility is still a factor, however, since HSH was not able to reach a feasible solution in 62 out of 160 instances in total.

Results for the computational requirements of the algorithms are presented in Tables 4.12-4.14. More specifically, Table 4.12 shows the average and maximum computational time needed by FMP to find the ideal value of each of the 6 criteria. Table 4.13 shows the total computational effort required by HSH for the associated single objective problems, which was computed as the sum of the computational times of the two 3-period subproblems. Table 4.14 shows the computational effort required by FMP and HSH for the weighted sums approach. Results for the computational performance of AFH1 and AFH2 are not reported, since their computational requirements are always negligible (less than a second). This justifies the effectiveness of these heuristics, especially for large FMP instances, for which the other available approaches are not applicable in reasonable times.

Table 4.12: Computational requirements (in seconds) needed by FMP to find the ideal value of each of the 6 criteria

|M|	|N_m|	z_1 Avg	z_1 Max	z_2 Avg	z_2 Max	z_3 Avg	z_3 Max	z_4 Avg	z_4 Max	z_5 Avg	z_5 Max	z_6 Avg	z_6 Max
2	**6**	0.024	0.050	0.040	0.124	0.332	0.759	0.326	0.814	0.018	0.022	2.756	20.181
2	**7**	0.069	0.269	0.085	0.299	3.588	32.027	0.430	1.829	0.037	0.123	99.997	519.030
2	**8**	0.069	0.336	0.062	0.175	0.187	0.704	0.471	1.169	0.038	0.186	557.701	1665.700
2	**9**	0.025	0.030	0.026	0.034	124.037	1232.71	0.410	0.791	0.023	0.028	696.355	1803.470
2	**10**	0.030	0.041	0.038	0.092	7.589	51.421	1.466	7.443	0.026	0.029	1171.418	1803.300
3	**6**	0.026	0.031	0.034	0.097	46.297	451.82	0.501	1.150	0.024	0.027	695.558	1803.910
3	**7**	0.032	0.042	0.040	0.073	10.474	83.351	1.004	1.636	0.030	0.033	1455.299	1804.290
3	**8**	0.036	0.044	0.048	0.144	5.236	23.413	1.829	9.106	0.033	0.037	1697.275	1803.610

Table 4.13: Computational requirements (in seconds) of HSH for each of the 6 single objective problems

|M|	|N_m|	z_1 Avg	z_1 Max	z_2 Avg	z_2 Max	z_3 Avg	z_3 Max	z_4 Avg	z_4 Max	z_5 Avg	z_5 Max	z_6 Avg	z_6 Max
2	**6**	0.014	0.019	0.016	0.030	0.029	0.055	0.029	0.050	0.013	0.016	0.038	0.101
2	**7**	0.015	0.019	0.016	0.029	0.026	0.040	0.028	0.045	0.013	0.017	0.057	0.165
2	**8**	0.016	0.018	0.017	0.025	0.027	0.036	2.314	16.010	0.014	0.016	0.170	0.249
2	**9**	0.016	0.019	0.016	0.019	0.071	0.219	1.918	17.011	0.014	0.017	0.127	0.328
2	**10**	0.017	0.021	0.020	0.024	0.270	0.751	2.287	20.014	0.016	0.018	0.303	0.764
3	**6**	0.018	0.023	0.016	0.022	0.077	0.136	0.030	0.051	0.016	0.021	0.188	0.561
3	**7**	0.021	0.026	0.026	0.050	0.588	3.957	2.380	21.047	0.020	0.026	1.130	5.303
3	**8**	0.024	0.028	0.024	0.028	0.272	0.899	0.044	0.074	0.022	0.025	1.006	4.396

Table 4.14: Computational requirements (in seconds) of FMP and HSH for the weighted sums approach

|M|	|N_m|	FMP, Z_A Avg	FMP, Z_A Max	HSH, Z_A Avg	HSH, Z_A Max	FMP, Z_B Avg	FMP, Z_B Max	HSH, Z_B Avg	HSH, Z_B Max
2	**6**	0.789	1.175	0.047	0.086	0.554	2.503	0.036	0.060
2	**7**	2.252	14.121	0.092	0.410	6.042	23.868	0.050	0.113
2	**8**	0.529	0.828	0.079	0.134	42.034	98.537	0.091	0.148
2	**9**	0.741	1.725	0.064	0.113	290.802	1803.650	0.102	0.242
2	**10**	2.511	10.819	0.102	0.174	944.721	1803.650	0.158	0.329
3	**6**	1.062	2.141	0.112	0.161	423.489	1804.050	0.211	0.443
3	**7**	6.447	31.741	0.134	0.247	1446.759	1804.610	0.435	1.215
3	**8**	9.050	41.949	0.196	0.281	1803.922	1804.550	0.589	1.123

A first observation that can be made based on the results of Tables 4.12-4.14 is that, besides problem size, the actual values of the problem parameters also have a strong influence on the total computational effort needed to reach an optimal solution in the case of FMP and HSH. This is supported by the fact that, even for the same problem size, a large variance is exhibited in the computational times of FMP and HSH. Note that the results of Tables 4.12-4.14 are only partly comparable, since HSH was not able to reach a feasible solution for all the problem instances that it was tested. On the other hand, the computational effort of AFH1 and AFH2 is negligible for all problem sizes. Our computational experience also indicates that the computational effort of AFH1 and AFH2 does not vary significantly even when the values of the problem parameters differ considerably. Therefore, the actual values of the problem parameters do not seem to have a strong influence on the computational effort of AFH1 and AFH2.

A careful comparison of the results presented in Tables 4.12 and 4.13 reveals that, in most cases, the total computational effort needed to reach an optimal solution decreases significantly when HSH is used instead of FMP, although not in all of them. While in the case of objectives z_1, z_2 and z_5 this decrease is not very large, it is much larger in the case of objectives z_3 and z_6. This is because the total computational effort is already small in the former case; therefore, there is little room for improving it. In the latter case, however, where the FMP computational effort is considerably larger, there exist significantly larger computational savings from utilizing HSH. In the case of z_4 on the other hand, the computational time seems to increase when HSH is used instead of FMP. This is a very important observation, because it shows that the proposed heuristics may behave differently than expected.

As far as the weighted sums approach is concerned, the results of Table 4.14 suggest that there are significant computational savings from utilizing HSH instead of FMP. A very interesting observation is that the computational effort needed for the application of the weighted sums approach with the Z_B objective on large problems becomes significantly larger than the computational effort needed for the application of the weighted sums approach with the Z_A objective.

Before concluding this section, it is noteworthy to make an additional important observation. In an attempt to directly compare an AFH with an FMP solution and discover the characteristics of the latter that make it advantageous, we discovered the following interesting FMP behavior: FMP rarely grounds an aircraft that will not exit the maintenance station by period T+1. The intuition behind this is trivial, since only the periods up to T+1 are included in the model's objective function; therefore, an aircraft exiting the maintenance station later than period T+1 cannot influence the objective of the current planning horizon. This strategy, however, may result in an excessive number of aircraft with particularly low residual flight times towards the end of current planning horizon, which in turn may result in a drastic decrease of the unit's fleet availability over the next planning horizon. The performance of AFH within a single time horizon can be similarly improved by only allowing the grounding of an aircraft if this aircraft is expected to exit the maintenance station by period T+1; this, however, may have an adverse effect on the unit's availability in the long term.

4.5 Summary

In this chapter, we addressed a multi-objective optimization model for the multi-period variant of the FMP problem. We proved several interesting theoretical properties for this problem, and we utilized them to develop two heuristics for solving this model. We also presented experimental results demonstrating the computational performance of these heuristics and the quality of the solutions that they produce. The results are very satisfactory, because they show that, under careful consideration, even large FMP instances can be handled quite effectively.

The first heuristic, AFH, is simple and performs quite satisfactorily in most of the cases. It is based on an ad-hoc heuristic technique used in many Air Force organizations worldwide. The question of whether the quality of the solutions that it produces can be further improved through appropriate enhancements remains open, leaving this as a potential direction for future research.

The second heuristic, HSH, exhibits a rather myopic behavior. It focuses on maximizing fleet availability in the initial periods first, which may result in low availability over the next periods. Nevertheless, the solution obtained by HSH is quite satisfactory in most cases. Therefore, it can be considered alternatively for obtaining a satisfactory solution when the size of the problem prohibits its solution using an exact solution algorithm. In general, the number of periods of each smaller horizon has a strong effect on the quality of the obtained solution by HSH. An interesting conclusion that arises from this observation is that, since this is an on-going problem repeatedly solved in successive horizons, the length of the horizons for which the wing command issues the flight requirements has a strong impact on the long term availability of the unit. As the number of periods over which the command issues the flight requirements increases, the fleet availability of the unit is expected to increase, too. This remark reveals the potential benefits from extending the planning horizon for which flight load information is available.

Chapter 5 Single Period FMP Problem

5.1 Introduction

The application of the aircraft flowchart presented in Section 3.3 involves two additional shortcomings that have an adverse critical effect on its efficiency. The first one stems from the fact that the slope of the diagonal varies depending on the exact number of aircraft that will be available in the next time period. The application of the underlying methodology, however, necessitates a diagonal whose slope is known in advance. Of course, this is not possible, since it requires knowledge of the flight and maintenance times of the individual aircraft in the current period, which are decision variables and not parameters. In practice, this difficulty is addressed by deciding in advance, based on simplistic heuristic rules, which aircraft will be available in the next time period and which not, but it becomes clear that these decisions are not necessarily optimal.

The second shortcoming of the aircraft flowchart methodology is that it does not provide a maintenance plan for the grounded aircraft. Several important questions pertaining to the grounded aircraft need to be answered to this end, such as the order in which they should receive service, the amount of service time that should be allocated to each individual aircraft, etc. In practice, such decisions are also made in a heuristic way, based on the intuition and the experience of the user.

In the previous chapter, we partially dealt with the above shortcomings, by providing a systematic methodology for utilizing the aircraft flowchart effectively. Our computational results demonstrate that, under careful consideration, the proposed heuristics can handle large FMP instances effectively, yielding satisfactory solutions in insignificant solution times. The optimality gaps, however, show that there is still room for improvement.

The problem's myopic nature adds another factor of difficulty that complicates its solution. More specifically, independently of the exact length of the planning horizon, finding the plan that provides the maximum readiness usually results in a drastic decrease of this readiness over subsequent horizons. This is an inevitable side effect, since focusing on one particular horizon in isolation and overlooking the requirements of the following ones does not result in long-term optimal decisions, but in short-term ones instead. Therefore, potential benefits can arise from extending the planning horizon for which flight load information is available. Our computational experience suggests that an efficient FMP model should ideally be able to provide solutions whose fleet availability exhibits low variability. This is mainly due to the fact that, since the FMP model is considered in subsequent rolling horizons in practice, the transition into the next planning horizon should always be as smooth as possible.

In order to address the aforementioned difficulties effectively, in this chapter we develop a mixed integer nonlinear model that can be used to generate the joint flight and maintenance plan of a unit of mission aircraft over a single-period planning horizon. This model aims to establish a balanced allocation of the flight load and the maintenance capacity to the individual aircraft of the unit, so that the long term availability is kept at a high and steady level. Its objective function minimizes a least squares index expressing the total deviation of the individual aircraft flight and maintenance times from their corresponding target values.

Using the model's special structure and properties, we develop a solution algorithm, which returns the exact optimum. We analyze the computational complexity of this algorithm and we present computational results comparing its performance against that of a commercial optimization package. Besides demonstrating the superiority of the proposed algorithm, these results reveal that the total computational effort required for the solution of the problem depends mainly on two crucial parameters: the size of the unit (i.e., the number of aircraft that comprise it), and the space capacity of the maintenance station.

5.2 Single period FMP Problem (S_{Per}-FMP)

For the mathematical formulation of the proposed optimization model, we introduce the following notation:

Decision Variables:

x_i : flight time of available aircraft i in the current time period,

h_j : maintenance time of grounded aircraft j in the current time period,

y_{in} : residual flight time of available aircraft i at the beginning of the next time period,

g_{jn} : residual maintenance time of grounded aircraft j at the beginning of the next time period,

b_i : binary decision variable that takes the value 1 if available aircraft i enters the maintenance station for service at the beginning of the next time period, and 0 otherwise,

c_j : binary decision variable that takes the value 1 if grounded aircraft j exits the maintenance station at the beginning of the next time period, and 0 otherwise,

z_g : number of aircraft that will enter the maintenance station at the beginning of the next time period,

z_a : number of aircraft that will exit the maintenance station at the beginning of the next time period.

Parameters:

S : required flight load in the current time period,

B : time capacity of the maintenance station in the current time period,

y_{ip} : residual flight time of available aircraft i at the beginning of the current time period,

g_{jp} : residual maintenance time of grounded aircraft j at the beginning of the current time period,

X_{max}: maximum flight time of any available aircraft in the current time period,

Y_{min} : lower bound on the residual flight time of every available aircraft,

G_{min} : lower bound on the residual maintenance time of every grounded aircraft,

C : maximum number of aircraft that the maintenance station can accommodate,

Y : residual flight time of an aircraft immediately after it exits the maintenance station,

G : residual maintenance time of an aircraft immediately after it enters the maintenance station,

L, U : real numbers denoting the maximum deviation from the target value of the flight load that can be tolerated,

A : number of available aircraft at the beginning of the current time period,

NA: number of grounded aircraft at the beginning of the current time period,

N : total number of aircraft in the unit = $A + NA$.

At the beginning of the current time period, the available aircraft are arranged in non-decreasing order of their residual flight times and the grounded aircraft are arranged in non-decreasing order of their residual maintenance times. We make the assumption that this order is always preserved, according to the aircraft flowchart methodology. Although this is not an actual restriction in practice, we will show in what follows that we can always assume that it is, without loss of generality. With this in mind, the problem under consideration is formulated as follows:

$$\underset{\substack{x_i, y_{in}, h_j, g_{jn}, \\ b_i, c_j, z_a, z_g}}{\text{Min}} \sum_{i=1}^{A}[(1-b_i)(y_{in}-(i-z_g)\frac{Y}{A-z_g+z_a})^2+b_i(G-(NA+i-z_a)\frac{G}{NA+z_g-z_a})^2]+ \sum_{j=1}^{NA}[(1-c_j)(g_{jn}-(j-z_a)\frac{G}{NA+z_g-z_a})^2+c_j(Y-(A+j-z_g)\frac{Y}{A-z_g+z_a})^2] \tag{5.2.1}$$

$$\text{s.t. } y_{in} = y_{ip} - x_i,\ i = 1,..,A \tag{5.2.2}$$

$$g_{jn} = g_{jp} - h_j,\ j = 1,..,NA \tag{5.2.3}$$

$$y_{in} \le y_{i+1n},\ i = 1,..,A\text{-}1 \tag{5.2.4}$$

$$g_{jn} \le g_{j+1n},\ j = 1,..,NA\text{-}1 \tag{5.2.5}$$

$$z_g = \sum_{i=1}^{A} b_i \tag{5.2.6}$$

$$z_a = \sum_{j=1}^{NA} c_j \tag{5.2.7}$$

$$NA + z_g - z_a \le C \tag{5.2.8}$$

$$\sum_{j=1}^{NA} h_j = \min(B, \sum_{j=1}^{NA} g_{jp}) \tag{5.2.9}$$

$$LS \le \sum_{i=1}^{A} x_i \le US \tag{5.2.10}$$

$$y_{in} \ge Y_{min}(1\text{-}b_i),\ i = 1,...,A \tag{5.2.11}$$

$$g_{jn} \ge G_{min}(1-c_j),\ j = 1,...,NA \tag{5.2.12}$$

$$y_{in} \le y_{ip}(1\text{-}b_i),\ i = 1,...,A \tag{5.2.13}$$

$$g_{jn} \le g_{jp}(1-c_j),\ j = 1,...,NA \tag{5.2.14}$$

$$x_i \le X_{max},\ i = 1,...,A \tag{5.2.15}$$

$$x_i \ge 0,\ y_{in} \ge 0,\ i = 1,..,A \tag{5.2.16}$$

$$h_j \ge 0,\ g_{jn} \ge 0,\ j = 1,..,NA \tag{5.2.17}$$

$$b_i \text{ binary},\ i = 1,..,A;\ c_j \text{ binary},\ j = 1,..,NA \tag{5.2.18}$$

$$z_g,\ z_a \text{ integer} \ge 0 \tag{5.2.19}$$

The objective function (5.2.1) minimizes the total deviation index that will be realized at the beginning of the next time period, i.e., the sum of squares of the deviations of the residual flight times of the available aircraft and the residual maintenance times of the grounded aircraft from their corresponding target values on the associated flowcharts. To see why this is true, note that this objective comprises of two summations. The first one refers to the aircraft which are available at the beginning of the current period. Consider a particular of these aircraft with index i. If $z_g < i$, then this aircraft will remain available in the next time period, too. Since the aircraft with the z_g smallest residual flight times will enter the maintenance station for service, the index of this aircraft on the flowchart of available aircraft at the beginning of the next time period will become equal to i-z_g. Additionally, since z_g aircraft will enter and z_a aircraft will exit the maintenance station, the new diagonal slope of this flowchart will become equal to $\frac{Y}{A - z_g + z_a}$. Thus, the corresponding target value of the residual flight time of this aircraft at the beginning of the next time period will be equal to $(i - z_g)\frac{Y}{A - z_g + z_a}$.

On the other hand, if $z_g \geq i$, then this aircraft will join the set of grounded aircraft at the beginning of the next time period with residual maintenance time equal to G, assuming an index j instead of i. Considering that z_a aircraft will exit the maintenance station at the beginning of the next period and that this will be the i^{th} in order aircraft to enter the maintenance station, the index of this aircraft on the flowchart of grounded aircraft at the beginning of the next time period will become equal to (NA+i–z_a). Moreover, the new diagonal slope of this flowchart will become equal to $\frac{G}{NA + z_g - z_a}$. Thus, the corresponding target value of the residual maintenance time of this aircraft at the beginning of the next time period will be equal to $(NA + i - z_a)\frac{G}{NA + z_g - z_a}$.

The terms (1-b_i) and b_i are binary indicators denoting whether aircraft i will be grounded or not in the next time period. More specifically, if $b_i = 0$, then aircraft i will retain its availability in the next period; therefore, the square of its residual flight time deviation on the flowchart of available aircraft will be taken into account in the objective function. On the other hand, if $b_i = 1$, then aircraft i will be grounded in the next period and the square of its residual maintenance time deviation on the flowchart of grounded aircraft will be taken into account in the objective function instead. The second summation of the objective function pertains, in an identical way, to the aircraft which are grounded at the beginning of the current period.

Constraint set (5.2.2) updates the residual flight time of each available aircraft based on its initial status and the flight time that will be assigned to it in the current time period, similarly to the constraint set (4.2.9) of the model presented in Section 4.2. Likewise, constraint set (5.2.3) updates the residual maintenance time of each grounded aircraft based on its initial status and the maintenance time that it will receive in the current time period similarly to the constraint set (4.2.12) of the model presented in Section 4.2. Constraints (5.2.4) and (5.2.5) ensure that the order of available aircraft and the order of grounded aircraft, respectively, will be preserved. Constraints (5.2.6) and (5.2.7) compute the number of aircraft that will enter and exit, respectively, the maintenance station at the beginning of the next time period, based on the values of binary variables b_i and c_j.

Constraint (5.2.8) ensures that the space capacity of the maintenance station will not be violated similarly to the constraint set (4.2.17) of the model presented in Section 4.2, while constraint (5.2.9) ensures that the total maintenance time that will be provided by the station will either be equal either to its total time capacity, or to the total maintenance requirements of the grounded aircraft, whichever of these two quantities is smaller. This constraint serves to ensure both that the time capacity of the maintenance station will not be violated, and that no part of this capacity will remain unused, whenever additional maintenance requirements exist. Constraint set (5.2.10) ensures that the required flight load will be satisfied, within a tolerance defined by variables L and U similarly to the constraint set (4.2.15) of the model presented in Section 4.2. For example, when L = 0.95 and U = 1.05, a maximum deviation of 5% from the target value of the flight load can be tolerated.

Constraint set (5.2.11) imposes a lower bound equal to Y_{min} on the residual flight time of each available aircraft that will remain available in the next period, too, similarly to the constraint set (4.2.27) of the model presented in Section 4.2. This bound is imposed when $b_i = 0$; otherwise, the corresponding constraint becomes redundant, since $b_i = 1$ implies that this aircraft will be grounded in the next period. This modeling technique prevents an available aircraft from ending up with positive but negligible residual flight time at the end of the current time period. Likewise, constraint set (5.2.12) imposes a lower bound of G_{min} on g_{jn} when $c_j = 0$, similarly to the constraint set (4.2.28) of the model presented in Section 4.2, and becomes redundant when $c_j = 1$. This prevents a grounded aircraft from ending up with positive but negligible residual maintenance time at the end of the current time period.

We introduce constraints (5.2.11) and (5.2.12) because it is odd and unrealistic to have a grounded (available) aircraft whose residual maintenance (flight) time is positive but arbitrarily small. This does not imply that a grounded aircraft whose residual maintenance time drops below G_{min} can be declared fit for flight and released from the maintenance station, since that would violate the safety standards. If an aircraft could indeed finish its service in a time frame which is strictly smaller than G, this would signify that the actual service duration is in fact less than G, and that G has been erroneously used. Moreover, if we allowed the solver to release an aircraft from the maintenance station before this aircraft finishes its service, then the solver would take advantage of this whenever it was preferable and would apply this smaller service time instead of G, in order to improve the quality of the returned solution. Similarly, imposing the lower bound Y_{min} on the residual flight time of each available aircraft does not imply that an available aircraft can be grounded, as soon as its residual flight time drops below Y_{min}, because this would increase the average maintenance cost per hour flown, decreasing in this way the long-term efficiency of the unit.

Constraint set (5.2.13) states that the residual flight time of an available aircraft must drop to 0 before this aircraft enters the maintenance station for service. This constraint is redundant when $b_i = 0$, and forces y_{in} to 0-value when $b_i = 1$, since it implies, together with constraint (5.2.2), that x_i will be equal to y_{ip}. Likewise, constraint set (5.2.14) states that the residual maintenance time of a grounded aircraft must drop to 0 before this aircraft exits the station and becomes available. This constraint is redundant when $c_j = 0$, and forces g_{jn} to 0-value when $c_j = 1$, as it implies, together with constraint (5.2.3), that h_j will be equal to g_{jp}. Constraint set (5.2.15) imposes an upper bound on the flight time of each available aircraft. This restriction is usually present due to technical reasons. Finally, constraints (5.2.16), (5.2.17) and (5.2.18), (5.2.19) are the non-negativity and the integrality constraints, respectively.

There exists a rare special case in which constraints (5.2.9), (5.2.12) and (5.2.14) cannot be satisfied simultaneously. This happens when both $B < \sum_{j=1}^{NA} g_{jp}$ and, at the same time, utilizing B fully by forcing $\sum_{j=1}^{NA} h_j$ to be equal to B (as required by constraint (5.2.9)) inevitably results in one or more grounded aircraft with residual maintenance time positive but strictly smaller than G_{min} at the end of the current period. For example, if $NA = 2$, $g_{1p} = g_{2p} = G_{min} = 0.1$ and $B = 0.05$, then the problem is infeasible. This is because there is no feasible way to utilize B fully without violating the restriction that the residual maintenance time of every grounded aircraft must be greater or equal to G_{min}.

In practice, a typical way to avoid leaving an aircraft with very small but positive residual maintenance time at the end of the current time period is by forcing the station to provide the extra time that this aircraft needs in order to finish its service. This implies that the time capacity of the station will be exogenously increased. Of course, if this extra time is negligible, this increase will also be negligible. In the small numerical example introduced above, for example, setting $B = 0.1$ would make the problem feasible with no significant effect.

From a mathematical point of view, on the other hand, a suitable modification of the problem formulation can be adopted to model this, as described next. Let t be an auxiliary binary decision variable that takes the value 1 if constraint (5.2.9) is relaxed and 0 otherwise. If $B < \sum_{j=1}^{NA} g_{jp}$, substitute constraint (5.2.9) with the following two constraints:

$$\sum_{j=1}^{NA} h_j \leq B + Ft \tag{5.2.20}$$

$$B \leq \sum_{j=1}^{NA} h_j \tag{5.2.21}$$

where F is a sufficiently large number, introduce the following constraint:

$$F(1-t) + \sum_{j=1}^{NA} h_j \geq \sum_{j=1}^{NA} g_{jp} - (NA - z_a)G_{min}, \tag{5.2.22}$$

and add the term Ft to the objective function, in order to ensure that variable t will always be set to 0 whenever a feasible solution to the original formulation exists. The insight of this modification is the following. If $B < \sum_{j=1}^{NA} g_{jp}$ and a feasible solution such that $\sum_{j=1}^{NA} h_j = B$ exists, then t will be set to 0 and this equality will be denoted by constraints (5.2.20) and (5.2.21), whereas constraint (5.2.22) will become redundant. Otherwise, t will be set to 1, constraint (5.2.20) will become redundant, and constraint (5.2.22) will set the total maintenance time that will be provided by the station equal to $\sum_{j=1}^{NA} g_{jp} - (NA - z_a)G_{min}$, instead of B. This is true even though constraint (5.2.22) is expressed as inequality instead of equality, since no feasible solution such that $\sum_{j=1}^{NA} h_j > \sum_{j=1}^{NA} g_{jp} - (NA - z_a)G_{min}$ will exist. Thus, it will be acceptable to slightly increase the station's time capacity in that case, under the restriction that every grounded aircraft that will not finish its service will end up with residual maintenance time equal to G_{min} at the end of the current time period. The user should keep in mind that if $t = 1$, the correct value of the total deviation index results after the term Ft is subtracted from the optimal objective function value.

When this modification in the problem formulation is adopted, the problem may be satisfied for several values of variable z_a. To ensure that the slight increase on the time capacity of the maintenance station will be the minimum possible, the user is advised to select the solution with the minimum feasible value for variable z_a. It seems logical that this will be the most desirable solution, since it will result in the smallest violation of the original constraint $\sum_{j=1}^{NA} h_j = B$. In the small numerical example introduced above, for example, a feasible solution can be obtained by setting $\sum_{j=1}^{NA} h_j$ equal to 0.1 and z_a

equal to 1, or by setting $\sum_{j=1}^{NA} h_j$ equal to 0.2 and z_a equal to 2. The former out of these two solutions seems preferable, since it results in the smallest violation of the original constraint $\sum_{j=1}^{NA} h_j = 0.05$.

A similar situation can arise when the satisfaction of the flight load inevitably results in one or more available aircraft with residual flight time strictly smaller than Y_{min} at the end of the current period. For example, if $A = 1$, $y_{1p} = Y_{min} = 0.1$, $S = 0.05$, $L = 0.9$ and $U = 1.1$, then the problem does not have a feasible solution. A modeling technique similar to the one presented above can be adopted in this case, too. Nevertheless, these two are rare extreme cases with practically negligible effect; therefore, in order to avoid complicating things unnecessarily, in what follows we assume that the problem defined by (5.2.1)-(5.2.19) is always feasible, and we suppress constraints (5.2.20)-(5.2.22).

The progressively worse deviation index for the subsequent aircraft that exit or enter the station besides the first one, and the fact that the model does not allow an aircraft to act both as available and as grounded within the same time period are main consequences of the planning horizon's discretization, which does not allow an aircraft to both fly and be serviced within the same time period. The minimization of the unit's total deviation index in subsequent time periods will balance the allocation of the flight and maintenance times and smoothen things out.

In practice, the flight load requirements refer to 1-month time periods, the time capacity of the maintenance station is measured in monthly labor hours and the flight/maintenance plans of the unit are typically reviewed and updated at the beginning of each month. This setting not only hints towards the development of a discrete model, but also implies that it is reasonable to choose an optimization period of one month, though it might seem preferable to make this choice in such a way that only one aircraft enters the station and only one aircraft exits the station at each time period.

This is not a very strict limitation for the present model, since the flight load issued by the unit command is practically small on the one hand, while the load factor of the maintenance station is practically equal to 1 on the other hand. In turn, the flight load of the unit can be satisfied using only few of the unit's aircraft, whereas the time capacity of the maintenance station is always fully utilized. Hence, in terms of the unit's long-term aggregate fleet availability, it is hardly ever an issue, if the flight time of one or more aircraft that are fit for flight is forced to zero value for an extra time period. Similarly, delaying the beginning of the maintenance service of one or more aircraft for an extra time period is not critical either, because other grounded aircraft that are present can always utilize the station's capacity fully. If it is absolutely necessary, the consequences of the progressively worse deviation index for the subsequent (besides the first one) aircraft that change status can be lessened by only including in the computation of the total deviation index the first aircraft that will become grounded and the first aircraft that will become available (i.e., by assuming a deviation of 0 for every subsequent aircraft that will change status).

The formulation that we propose adopts the aircraft flowchart methodology both for the available and for the grounded aircraft. Its novelty lies in that it gives the user additional flexibility by allowing the slopes of the two diagonals to vary, based on the actual number of aircraft that enter and exit the maintenance station. This seems far more rational than fixing these numbers in advance and generating the flight and maintenance plans afterward, based on the resulting diagonal slopes.

We can essentially view this formulation as the composition of two individual ones, one that pertains to the available and one that pertains to the grounded aircraft. The main difference between them is that a small deviation, determined by variables L and U, can be tolerated for the satisfaction of the flight load by the available aircraft, whereas no such tolerance exists for the total maintenance time that will be provided by the maintenance station. Additionally, an upper bound is imposed on the maximum flight time of each available aircraft, whereas no such bound is imposed on the maintenance time of each grounded aircraft.

At first glance, the problem formulation seems too restricting, since it imposes a steady rotation of the aircraft in and out of the maintenance station in non-decreasing order of their residual flight/maintenance times, without allowing any contravention of this order. For example, according to this formulation, an aircraft which is available in the current period cannot be grounded at the beginning of the next one, unless all the available aircraft with smaller residual flight times are grounded, too. Similarly, the grounded aircraft must always exit the maintenance station in the exact same order in which they entered it in the first place.

In the actual application that we study, no such restriction is present. Aircraft are allowed to enter and exit the maintenance in any feasible order, while their indices are updated accordingly to represent their relative order in terms of their residual flight and maintenance times. With this in mind, the index of each aircraft at the beginning of the next period should be a decision variable allowed to take any feasible value and should not be determined by the exact number of aircraft that will enter and exit the maintenance station. Nevertheless, we prove next that there always exists at least one optimal solution to the problem that results when this restriction is relaxed that satisfies it. Besides establishing the validity of the proposed formulation, this proof also reveals some crucial properties of the problem, which are utilized in the next section for the development of the exact solution algorithm.

Proposition 5.1: Given the optimal solution to the problem defined by (5.2.1)-(5.2.19), there does not exist another solution that satisfies all the constraints except possibly (5.2.4) and/or (5.2.5), in which the two flowcharts of available and grounded aircraft at the beginning of the next time period result in lower total deviation index.

Proof. Suppose that there exists another feasible solution (let the superscript q refer to the values of the decision variables of that solution) that contains two initially available aircraft with indices l and m, such that $y_{lp} > y_{mp}$ and $y_{ln}^q < y_{mn}^q$, and results in lower total deviation index. Consider another solution in which all the decision variables take the same values as those of solution q, except that $x_l = (y_{lp} - y_{mp}) + x_m^q$, $x_m = x_l^q + x_m^q - x_l$, $y_{ln} = y_{mn}^q$, $y_{mn} = y_{ln}^q$, $b_l = b_m^q$ and $b_m = b_l^q$. It is easy to verify that aircraft l and m no longer violate constraint set (5.2.4). Moreover, if there is a one to one interchange of these aircraft and the relative order of the remaining aircraft on the associated flowcharts is kept the same, the total deviation index value of this solution is the same as that of solution q. Repeating this procedure for any two initially available aircraft that violate constraint set (5.2.4) in solution q, we can eventually get a solution that has the same total deviation index value and satisfies constraint set (5.2.4) entirely. An identical procedure (just exchange variables x, y and b in the above substitution with the corresponding variables h, g and c, respectively) can also be applied for any two initially grounded aircraft that violate constraint set (5.2.5), leading eventually to a solution in which constraint set (5.2.5) is entirely satisfied, too. Moreover, the total deviation index value of this solution will be the same as that of solution q. This implies, however, that this solution provides an improvement to the objective function value of the optimal solution to the problem formed by expressions (5.2.1)-(5.2.19), contradicting its optimality.

5.3 Solution methodology

5.3.1 Solving for a particular combination of z_g and z_a

Suppose that the optimal values of variables z_g and z_a in the formulation (5.2.1)-(5.2.19) are known. Then, the optimal values of variables b_i for $i = 1,...,A$ and c_j for $j = 1,...,NA$, the slopes of the flowchart diagonals at the beginning of the next period and the two sets of aircraft that will be available and grounded at the beginning of the next period are known, too. In this case, obtaining the optimal values of the remaining decision variables reduces to solving two independent trivial subproblems, as explained next. The first one of these subproblems pertains to the set of aircraft that will be available at the beginning of the next period, whereas the second one pertains to the set of aircraft that will be grounded.

Let K denote the former of these sets, indexed by k. Set K is the union of set K_1, which consists of the aircraft which are grounded at the beginning of the current period and will exit the maintenance station at the beginning of the next one, and set K_2, which consists of the aircraft which are available at the beginning of the current period and will remain available in the next period, too. Clearly, the maintenance time of each aircraft in set K_1 must be set equal to its residual maintenance time at the beginning of the current period.

Similarly, let M denote the latter of these sets, indexed by m. Set M is the union of set M_1, which consists of the aircraft which are available at the beginning of the current period and will be grounded at the beginning of the next one, and set M_2, which consists of the aircraft which are grounded at the beginning of the current period and will remain grounded in the next period, too. Clearly, the flight time of each aircraft in set M_1 must be set equal to its residual flight time at the beginning of the current period.

We update the indices of the aircraft of set K as follows. The aircraft of set K_2 are indexed first with indices $k = 1,\ldots,|K_2|$ according to their residual flight time order, followed by the aircraft of set K_1 with indices $k = |K_2|+1,\ldots,|K_2|+|K_1|$, according to their residual maintenance time order. Noting that $|K| = |K_2|+|K_1|$, the following quadratic optimization problem can be used to find the optimal flight times of the aircraft that comprise set K_2:

$$\min_{x_k, y_{kn}} \sum_{k=1}^{|K|} \left(y_{kn} - k\frac{Y}{A - z_g + z_a}\right)^2$$

$$\begin{aligned}
\text{s.t.} \quad & y_{kn} = y_{kp} - x_k, \; k = 1,\ldots,|K_2| \\
& y_{kn} = Y, \; k = |K_2|+1,\ldots,|K| \\
& LS \leq \sum_{k=1}^{|K_2|} x_k + \sum_{m \in M_1} y_{mp} \leq US \\
& y_{kn} \geq Y_{min}, \; k = 1,\ldots,|K_2| \\
& x_k \leq X_{max}, \; k = 1,\ldots,|K_2| \\
& x_k \geq 0, \; k = 1,\ldots, |K_2|
\end{aligned}$$

In this formulation, the objective function minimizes the total deviation index that will be realized on the flowchart of available aircraft at the beginning of the next period. The first two sets of constraints update the residual flight times of the aircraft of sets K_2 and K_1, respectively, at the beginning of the next time period. The next constraint ensures that the flight requirements of the current period will be satisfied (index m scans the available aircraft that will use up their entire residual flight time and will enter the maintenance station for service at the beginning of the next period). The next two sets of constraints impose a lower bound on the residual flight time of each aircraft in set K_2 at the beginning of the next period, and an upper bound on the flight time of each aircraft in set K_2, respectively. Finally, the last set of constraints accounts for the non-negativity of the flight times. Note that the y_{kn}'s and the x_k's for $k = 1,\ldots,|K_2|$ are decision variables in this formulation, the y_{kp}'s for $k = 1,\ldots,|K_2|$ are known parameters, and the y_{kn}'s for $k = |K_2|+1,\ldots,|K|$ are auxiliary decision variables (they do not appear in the original formulation) with known values. Adding the auxiliary decision variables x_k for k = $|K_2|+1,\ldots,|K|$ and setting $s = \frac{Y}{A - z_g + z_a}$, y_{kp} = Y for k = $|K_2|+1,\ldots,|K|$, $X_k = \min(X_{max}, y_{kp} - Y_{min})$ for $k = 1,\ldots,|K_2|$, and $X_k = 0$ for $k = |K_2|+1,\ldots,|K|$, we obtain the following equivalent formulation:

$$\underset{x_k}{\text{Min}} \sum_{k=1}^{|K|} ((y_{kp} - x_k) - ks)^2 \tag{5.3.1}$$

$$\text{s.t. } LS - \sum_{m \in M_1} y_{mp} \le \sum_{k=1}^{|K|} x_k \tag{5.3.2}$$

$$\sum_{k=1}^{|K|} x_k \le US - \sum_{m \in M_1} y_{mp} \tag{5.3.3}$$

$$0 \le x_k \le X_k,\ k = 1,\ldots,|K| \tag{5.3.4}$$

The problem defined by (5.3.1)-(5.3.4) is a quadratic program equivalent to the quadratic program defined by (4.3.1)-(4.3.4). Following the rationale of Section 4.3, we utilize the following crucial and interesting result in the development of the proposed solution algorithm:

Proposition 5.2. Assume that the quantities LL, UL, X and D are arranged in non-decreasing order.
a) If, after taking into consideration any ties present, there does not exist an arrangement in which LL precedes X, then the problem defined by (5.3.1)-(5.3.4) is infeasible.
b) If an arrangement in which LL precedes X exists, then the optimal solution of the problem defined by (5.3.1)-(5.3.4) is the one obtained by Procedure Sweep when the sum of the assigned aircraft flight times becomes equal to the quantity that appears in the second place of this arrangement.
Proof. Same as Proposition 4.1 (See Appendix A).

We update the indices of the aircraft of set M as follows. The aircraft of set M_2 are indexed first with indices $m = 1,\ldots,|M_2|$ according to their residual maintenance time order, followed by the aircraft of set M_1 with indices $m = |M_2|+1,\ldots,|M_2|+|M_1|$, according to their residual flight time order. Noting that $|M| = |M_2|+|M_1|$, the following quadratic optimization problem can be used to find the optimal maintenance times of the aircraft that comprise set M_2:

$$\underset{h_m, g_{mn}}{\text{Min}} \sum_{m=1}^{|M|} \left(g_{mn} - m\frac{G}{NA + z_g - z_a}\right)^2$$

$$\text{s.t. } g_{mn} = g_{mp} - h_m,\ m = 1,\ldots,|M_2|$$

$$g_{mn} = G,\ m = |M_2|+1,\ldots,|M|$$

$$\sum_{m=1}^{|M_2|} h_m + \sum_{k \in K_1} g_{kp} = \min\left(B, \sum_{j=1}^{NA} g_{jp}\right)$$

$$g_{mn} \ge G_{min},\ m = 1,\ldots,|M_2|$$

$$h_m \ge 0,\ m = 1,\ldots,|M_2|$$

In this formulation, the objective function minimizes the total deviation index that will be realized on the flowchart of grounded aircraft at the beginning of the next period. The first two sets of constraints update the residual maintenance times of the aircraft of sets M_2 and M_1, respectively, at the beginning of the next time period. The next constraint ensures that the time capacity of the maintenance station will be properly utilized (index k scans the grounded aircraft that will finish their service and will exit the maintenance station at the beginning of the next period). The next set of constraints imposes a lower bound on the residual flight time of each grounded aircraft in set M_2 at the beginning of the next period. Finally, the last set of constraints accounts for the non-negativity of the maintenance times. Note that the g_{mn}'s and the h_m's for $m = 1,\ldots,|M_2|$ are decision variables in this formulation, the g_{mp}'s for $m = 1,\ldots,|M_2|$ are known parameters, and the g_{mn}'s for $m = |M_2|+1,\ldots,|M|$ are auxiliary decision variables (they do not appear in the original formulation) with known values. Adding the auxiliary decision variables h_m for $m = |M_2|+1,\ldots,|M|$ and setting $s = \dfrac{G}{NA + z_g - z_a}$, $g_{mp} = G$ for $m = |M_2|+1,\ldots,|M|$, $H_m = g_{mp} - G_{min}$ for $m = 1,\ldots,|M_2|$, and $H_m = 0$ for $m = |M_2|+1,\ldots,|M|$, we obtain the following equivalent formulation:

$$\min_{h_m} \sum_{m=1}^{|M|} ((g_{mp} - h_m) - ms)^2 \tag{5.3.5}$$

$$\text{s.t. } \sum_{m=1}^{|M|} h_m = \min(B, \sum_{j=1}^{NA} g_{jp}) - \sum_{k \in K_1} g_{kp} \tag{5.3.6}$$

$$0 \le h_m \le H_m,\ m = 1,\ldots,|M| \tag{5.3.7}$$

This problem is similar to the one defined by (5.3.1)-(5.3.4). On the flowchart that results from the known values of parameters g_{mp} in (5.3.5)-(5.3.7), consider the following two solutions that can be obtained during the application of Procedure Sweep:

1. The solution in which the sum of the assigned aircraft maintenance times is equal to $\min(B, \sum_{j=1}^{NA} g_{jp}) - \sum_{k \in K_1} g_{kp}$. In what follows, we refer with "*Bg*" to this sum.

2. The solution in which each aircraft, m, is assigned its maximum possible maintenance time, H_m. In what follows, we refer with "*H*" to the sum of the assigned aircraft maintenance times of this solution. Then, Proposition 5.2 can be modified as follows, in order to identify the optimal solution of the problem defined by (5.3.5)-(5.3.7):

Proposition 5.3. If $Bg \leq H$, then the optimal solution of the problem defined by (5.3.5)-(5.3.7) is the one obtained by Procedure Sweep when the sum of the assigned aircraft maintenance times becomes equal to Bg. If $Bg > H$, then the problem defined by (5.3.5)-(5.3.7) is infeasible.

Proof. The validity of this proposition results from the fact that the problem defined by (5.3.5)-(5.3.7) is a special case of the problem defined by (5.3.1)-(5.3.4) with $L = U = 1$ and $LS - \sum_{m \in M_1} y_{mp} = US - \sum_{m \in M_1} y_{mp}$.

5.3.2 The general case

The above discussion implies that when the optimal values of variables z_g and z_a are known, obtaining the optimal values of the remaining decision variables of problem (5.2.1)-(5.2.19) reduces to solving two independent trivial subproblems. The first one of these subproblems is associated with the flowchart of the aircraft that will be available at the beginning of the next time period, while the second one is associated with the flowchart of the aircraft that will be grounded at the beginning of the next time period. Moreover, the optimal total index deviation value of the problem for a particular value combination of z_g and z_a is equal to the sum of the optimal deviation index values of these two subproblems. Therefore, among all the feasible value combinations of variables z_g and z_a, the problem's global optimal solution is associated with the one for which the total deviation index assumes its lowest value.

In order to exclude in advance infeasible value combinations of variables z_g and z_a, we consider variable z_a first. Assume that the grounded aircraft are already arranged in non-decreasing order of their residual maintenance times, with indices $j = 1,\ldots,NA$. Let *Sum* be a non-negative auxiliary variable. Then, the following pseudocode provides a valid upper bound on variable z_a:

```
Sum = 0; j = 1;
while j ≤ NA do
        Sum = Sum + g_jp;
        if Sum ≤ B do
                j = j + 1;
        else
                print j-1 and exit;
        end if
end while
print j-1;
```

Proposition 5.4. The value of variable z_a in any feasible solution to problem (5.2.1)-(5.2.19) cannot be larger than the value printed by the above pseudocode.

Proof. We use variable *Sum* to store the sum of the residual maintenance times of the grounded aircraft that can finish their service in the current time period. Every time that the residual maintenance time of an aircraft is added to *Sum* and *Sum* still remains less than or equal to *B*, we conclude that the time capacity of the maintenance station suffices to finish the service of that aircraft, too; therefore, we increase the upper bound on variable z_a by 1. The procedure terminates either when *Sum* becomes strictly greater than B, implying that the station's time capacity is not sufficient to finish the service of the last (or any subsequent) considered aircraft, or when the full list of grounded aircraft has been scanned, implying that NA is the maximum (and unique, due to constraint (5.2.9)) feasible value of variable z_a.

The following pseudocode provides a valid lower bound on variable z_a:

```
Sum = Σ_{j=1}^{NA} g_jp − NA(G_min); j = 0;
while j ≤ NA do
        if Sum ≥ B do
                print j and exit;
        else
                Sum = Sum + G_min; j = j + 1;
        end if
end while
print NA;
```

Proposition 5.5. The value of variable z_a in any feasible solution to problem (5.2.1)-(5.2.19) cannot be smaller than the value printed by the above pseudocode.

Proof. The validity of the proposition results from the fact that if k is a feasible value for variable z_a, then $\sum_{j=1}^{k} g_{jp} + \sum_{j=k+1}^{NA} (g_{jp} - G_{min}) \geq B$, or equivalently, $\sum_{j=1}^{NA} g_{jp} - NA(G_{min}) + k(G_{min}) \geq B$ holds. Thus, the lowest non-negative integer value of k for which this inequality holds is a valid lower bound on variable z_a. If no value of k between 0 and NA satisfies this condition, then $\sum_{j=1}^{NA} g_{jp} < B$, and constraint (5.2.9) restricts $\sum_{j=1}^{NA} h_j$ to be equal to $\sum_{j=1}^{NA} g_{jp}$; therefore, the only feasible value of variable z_a is NA.

In the presence of the special case discussed in Section 5.2 that we resolve through the introduction of constraints (5.3.1)-(5.3.3), the lower bound on variable z_a will be larger than its upper bound, rendering the entire problem infeasible. Next, we exclude infeasible values of variable z_g. Assume that the available aircraft are already arranged in non-decreasing order of their residual flight times, with indices $i = 1,\ldots,A$. The following pseudocode provides a valid upper bound on variable z_g:

```
Sum = 0; i = 1;
while i ≤ min(C,A) and y_ip ≤ X_max do
        Sum = Sum + y_ip;
        if Sum ≤ US do
                i = i + 1;
        else
                print i-1 and exit;
        end if
end while
print i-1;
```

Proposition 5.6. The value of variable z_g in any feasible solution to problem (5.2.1)-(5.2.19) cannot be larger than the value printed by the above pseudocode.

Proof. We use variable *Sum* to store the sum of the residual flight times of the available aircraft that can be grounded in the current time period. Every time that the residual flight time of an aircraft is added to *Sum* and *Sum* still remains less than or equal to *US*, we conclude that the flight load suffices to ground this aircraft, too; therefore, we increase the upper bound on variable z_g by 1. The procedure terminates either when the upper bound on variable z_g assumes its largest possible value (= min(*C*,*A*)), or when the first aircraft with residual flight time strictly greater than X_{max} is encountered, or when *Sum* becomes strictly greater than *US*.

The following pseudocode provides a valid lower bound on variable z_g:

```
Sum = Σ_{j=1}^{A} y_ip − A(Y_min); j = 0;
while j ≤ A do
        if Sum ≥ LS do
                print j and exit;
        else
                Sum = Sum + Y_min; j = j + 1;
        end if
end while
print "problem is infeasible";
```

Proposition 5.7. The value of variable z_g in any feasible solution to problem (5.2.1)-(5.2.19) cannot be smaller than the value printed by the above pseudocode.

Proof. The validity of the proposition results from the fact that if *k* is a feasible value for variable z_g then $\sum_{i=1}^{k} y_{ip} + \sum_{i=k+1}^{A} (y_{ip} - Y_{min}) \geq LS$, or equivalently, $\sum_{i=1}^{A} y_{ip} - A(Y_{min}) + k(Y_{min}) \geq LS$ holds. Thus, the lowest non-negative integer value of *k* for which this inequality holds is a valid lower bound on variable z_g. If this inequality does not hold even when $k = A$, then the problem is clearly infeasible.

Once we determine the individual upper and lower bounds on variables z_g and z_a, we utilize constraint (5.2.8) to eliminate those value pairs that violate the space capacity of the maintenance station. After that, the proposed algorithm searches among the remaining value combinations of variables z_g and z_a to find the one that leads to the lowest total deviation index.

5.3.3 A small numerical example

In this section, we illustrate the application of the above algorithm on a small numerical example. Consider a unit comprising of 6 aircraft, 4 of which are available and 2 of which are grounded at the beginning of the current time period. Table 5.1 presents the residual flight times of the available aircraft and the residual maintenance times of the grounded aircraft. In this table, bold-style entries denote maintenance times of grounded aircraft and plain-style entries denote flight times of available aircraft.

Table 5.1: Residual flight/**maintenance** times (y_{ip}/$\mathbf{g_{jp}}$) (hours)

$i = 1$	$i = 2$	$i = 3$	$i = 4$	$\mathbf{j = 1}$	$\mathbf{j = 2}$
38	50	273	298	**130**	**300**

The values of the other problem parameters are $S = 125$ hours, $B = 325$ hours, $G = 320$ hours, $Y = 300$ hours, $C = 3$, $X_{max} = 50$ hours, $Y_{min} = 0.1$ hours, $G_{min} = 0.1$ hours, $L = 0.95$ and $U = 1.05$.

The available/grounded aircraft are already sorted in non-decreasing order of their residual flight/maintenance times. If this were not the case, the user would have to rearrange them and update their indices accordingly. Applying the procedure for obtaining the bounds on variables z_g and z_a, we get $0 \leq z_g \leq 2$ and $0 \leq z_a \leq 1$. Only 5 of the resulting 6 value combinations of z_g and z_a are feasible, since constraint (5.2.8) is violated for $z_g = 2$ and $z_a = 0$. Table 5.2 presents the total deviation index value of each combination.

Table 5.2: Optimal total deviation index value for each combination of z_g and z_a

$z_g \setminus z_a$	**0**	**1**
0	87652.08	23098.56
1	27215.81	**9864.06**
2	IF	37819.62

Thus, the optimal solution is the one with $z_g = 1$ and $z_a = 1$ and objective function value 9864.06. This total deviation index value results as follows. Since $z_g = 1$ and $z_a = 1$, the available aircraft with index $i = 1$ will be grounded and the grounded aircraft with index $j = 1$ will become available at the beginning of the next time period. Therefore, $x_1 = 38$, $y_{1n} = 0$, $h_1 = 130$ and $g_{1n} = 0$.

Set K comprises of the available aircraft with initial indices $i = 2$, 3 and 4 (these aircraft comprise set K_2) and the grounded aircraft with initial index $j = 1$ (this aircraft comprises set K_1). Set M comprises of the grounded aircraft with initial index $j = 2$ (this aircraft comprises set M_2) and the available aircraft with initial index $i = 1$ (this aircraft comprises set M_1). In order to compute the values of the remaining decision variables, we update the aircraft indices first, as shown in Table 5.3.

Table 5.3: Update of the aircraft indices

Initial index	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$j = 1$	$j = 2$
Updated index	$j = 2$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$j = 1$

Using the updated indices, the following quadratic optimization problem can be used to compute the flight times of the aircraft that comprise set K_2:

$$\underset{y_{in},\, x_i}{\text{Min}}(y_{1n} - \frac{300}{4-1+1})^2 + (y_{2n} - 2\frac{300}{4-1+1})^2 + (y_{3n} - 3\frac{300}{4-1+1})^2 + (y_{4n} - 4\frac{300}{4-1+1})^2$$

$$\text{s.t.} \quad y_{1n} = 50 - x_1$$
$$y_{2n} = 273 - x_2$$
$$y_{3n} = 298 - x_3$$
$$y_{4n} = 300 - x_4$$

$$0.95(125) \leq x_1 + x_2 + x_3 + x_4 + 38 \leq 1.05(125)$$

$$y_{in} \geq 0.1,\ i = 1,\ldots,3$$
$$x_i \leq 50,\ i = 1,\ldots,3$$
$$x_4 = 0$$
$$x_i \geq 0,\ i = 1,\ldots,4$$

After some basic manipulation, we obtain the following equivalent formulation:

$$\underset{x_i}{\text{Min}}((50 - x_1) - 75)^2 + ((273 - x_2) - 150)^2 + ((298 - x_3) - 225)^2 + ((300 - x_4) - 300)^2$$

$$\text{s.t. } 80.75 \leq x_1 + x_2 + x_3 + x_4 \leq 93.25$$

$$0 \leq x_1 \leq 49.9;\ 0 \leq x_2 \leq 50;\ 0 \leq x_3 \leq 50;\ 0 \leq x_4 \leq 0$$

Arranging the quantities *LL*, *UL*, *X* and *D* of Proposition 5.2 in non-decreasing order, we have: $LL = 80.75$, $UL = 93.25$, $D = 100$, $X = 149.9$. Therefore, the optimal solution of the problem is the one obtained by Procedure Sweep when the sum of the assigned aircraft flight times becomes equal to 93.25. Using the updated indices, the flight times of the aircraft of set K_2 in that solution are $x_1 = 0$, $x_2 = 50$ and $x_3 = 43.25$, or, using the initial indices, $x_2 = 0$, $x_3 = 50$ and $x_4 = 43.25$. After substituting these values in the objective function of the above formulation, the deviation index value on the flowchart of the aircraft that will be available at the beginning of the next time period is equal to 6839.06. Similarly, the optimal maintenance times are 130 for the grounded aircraft with initial index $j = 1$ and 195 for the grounded aircraft with initial index $j = 2$, leading to a deviation index value of 3025 on the flowchart of the aircraft that will be grounded at the beginning of the next time period. Therefore, the total deviation index of the solution with $z_g = 1$ and $z_a = 1$ is equal to 6839.06 + 3025 = 9864.06.

5.4 Computational implementation

In this section, we analyze the worst-case computational complexity of the proposed algorithm, and we present computational results evaluating its performance on randomly generated instances.

5.4.1 Computational complexity

Problems similar to the one defined by (5.3.1)-(5.3.4) have been studied extensively in the past (Helgason et al. (1980); Brucker (1984); Calamai and Moré (1987); Pardalos and Kovoor (1990)) and several exact solution algorithms have been proposed, some of which are asymptotically optimal in terms of computational performance. Next, we utilize one of these papers in order to analyze the computational effort required to solve the problem defined by (5.3.1)-(5.3.4).

Lemma 5.1. The problem defined by (5.3.1)-(5.3.4) can be solved in time O(|*K*|).
Proof. Same as Lemma 4.1 (See Appendix B).

The following proposition utilizes Lemma 5.1 in order to analyze the computational complexity of the proposed solution algorithm.
Proposition 5.8. The computational complexity of the solution algorithm that we propose is O(*A* log(*A*)) + O(*N*((*NA*)(min(*C*,*A*)))).
Proof. The total time required to arrange the available aircraft in non-decreasing order of their residual flight times is O(*A* log(*A*)). The total time required to arrange the grounded aircraft in non-decreasing order of their residual maintenance times is O(*NA* log(*NA*)). The total time required to find the upper and lower bound on variable z_g is O(*A*). The total time required to find the upper and lower bound on variable z_a is O(*NA*). The total time required to find the value combinations of variables z_g and z_a that satisfy constraint (5.2.8) is O((*NA*)(min(*C*,*A*))), since z_a cannot have more than *NA*+1 feasible values and z_g cannot have more than min(*C*+1,*A*+1) feasible values. Solving the problem for a particular value combination of z_g and z_a requires total time O(*A*) + O(*NA*) = O(*A*+*NA*) = O(*N*). Since there are at most O((*NA*)(min(*C*,*A*))) such combinations, the total computational complexity of the proposed solution algorithm is O(*A* log(*A*)) + O(*NA* log(*NA*)) + O(*A*) + O(*NA*) + O((*NA*)(min(*C*,*A*))) + O((*N*)((*NA*)(min(*C*,*A*)))) = O(*A* log(*A*)) + O(*N*((*NA*)(min(*C*,*A*)))).

5.4.2 Computational results

We implemented the proposed solution algorithm in C/C++ and we compared its performance against that of two models that we developed in LINGO 11.0 [8]. LINGO is a commercial optimization software that can be alternatively utilized for the solution of the problem under consideration. We performed our computational experiments on an i7-920 @ 2.7 GHz Intel processor with 3 GB system memory. We used 5 different values (i.e., 50, 100, 150, 200 and 250) for the total number of aircraft that comprise the unit, and solved 20 random problem instances for each of them.

The first LINGO model that we developed (called *original* hereafter) solves the original problem formulation of Section 5.2, whereas the second one (called *decoupling* hereafter) utilizes the results of Section 5.3.2 to find the feasible value pairs of variables z_g and z_a, and then searches among them to find the one that results in the lowest total deviation index. This is doable, because LINGO has embedded capabilities that allow it to function as a programming language. The application of the decoupling LINGO model resembles the application of our specialized algorithm, the only difference being that it does not utilize Procedure Sweep to solve each of the smaller subproblems, but its own nonlinear programming subroutines.

We invoked LINGO mainly with default options, except that we modified the following options in order to improve its performance: a) we increased the *Update Interval* in the *Solver Status Window* from 2 to 60 seconds to prevent LINGO from spending too much updating this window's information, b) we set the *Output Level* at the *Interface Tab* to *Terse* to suppress useless output, c) we imposed the maximum of *Y*, *G* and *C* as an (obvious) upper bound on the optimal value of each decision variable in the *Variable Upper Bound* box of the *Global Server* tab.

We generated the random problem instances as follows: We set parameter *C* equal to 0.2*N*, rounded up to the nearest integer. Although *C* is equal to approximately 0.1*N* in practice (for a group of 60-80 aircraft, the maintenance hangar can typically accommodate 6-8 aircraft), we doubled this value in our design to make the generated problem instances more challenging. This is because according to the analysis of Section 5.4.1, as the value of *C* increases, the computational complexity of the proposed solution algorithm increases, too. We generated the number of grounded aircraft, *NA*, randomly, using a discrete uniform probability function that considered integer values between 0.1 *N* and 0.2 *N*, inclusive. Of course, we always set the number of available aircraft, *A*, equal to *N*-*NA*. The residual flight time of each available aircraft was a random number distributed uniformly in the interval $[Y_{min},Y]$ and the residual maintenance time of each grounded aircraft was a random number distributed uniformly in the interval $[G_{min},G]$.

We set parameter *B* equal to $0.75\sum_{j=1}^{NA} g_{jp}$ in order to maximize the number of feasible values for variable z_a, since only very large values are feasible for z_a as *B* approaches $\sum_{j=1}^{NA} g_{jp}$, and only very small as it approaches 0. Similarly, we set parameter *S* equal to $0.5\sum_{i=1}^{A} \min(y_{ip}, X_{\max})$, since only very large values are feasible for z_g as S approaches $\sum_{i=1}^{A} \min(y_{ip}, X_{\max})$, and only very small as it approaches 0. The difference in the multiplying coefficient (0.75 vs. 0.5) is partially due to the fact that a small deviation can be tolerated for the satisfaction of the flight load by the available aircraft, while no such tolerance exists for the total maintenance time that will be provided by the maintenance station. We used actual values drawn from the real application for the other problem parameters, i.e., $Y = 300$, $G = 320$, $L = 0.95$, $U = 1.05$, $X_{max} = 50$, $Y_{min} = 0.1$ and $G_{min} = 0.1$.

In order to test the proposed solution algorithm on large scale problems, too, we also applied it on problem instances with $N = 500$, 1000, 1500, 2000 and 2500. We were not able to apply either of the two LINGO models on these problems, since their computational requirements are prohibitive even when LINGO's local solver is invoked. Typical combat wings of the HAF may consist of up to 100 aircraft; therefore, it seems highly unlikely that problems of this magnitude will need to be solved in practice. Note, however, that for the needs of providing a plan over a typical planning horizon, the underlying problem may have to be solved repeatedly a significant number of times under possibly different scenarios. Therefore, a high speed solution algorithm, such as the one that we propose, is essentially important and will provide the additional capability of performing more thorough analyses and comparisons.

Tables 5.4 and 5.5 present the results of our computational experiments. Table 5.4 shows the computational requirements of the two LINGO models and of the solution algorithm that we propose. More specifically, columns 2 and 3 of this table show the average and maximum computational times when the original LINGO model with the global solver was invoked, whereas the next two columns show the average and maximum computational times when the original LINGO model with the local solver was invoked instead (the option *Use Global Server* in the *Global Solver* tab was unchecked in this case). The next two columns show the average and maximum computational times for the decoupling LINGO model. This model was only applied once to each random problem, since each of the smaller subproblems that arise when the values of z_g and z_a are known is a convex quadratic program whose global optimal solution can successfully be provided by both the global and the local solver of LINGO. Naturally, the global solver needs significantly more time than the local solver to identify this optimal solution; therefore, this model was only applied with the local solver invoked. Only results with up to $N = 200$ aircraft are reported for this model, since its computational requirements increase faster than those of the original LINGO model and become extraordinary for $N \geq 250$. Columns 8-9 of Table 5.4 present our proposed solution algorithm's average and maximum computational times. We explain the results shown in the last two columns of Table 5.4 in the next subsection.

Table 5.4: Computational requirements (in seconds) of the proposed solution algorithms and LINGO

	Original LINGO				Decoupling LINGO		Proposed algorithm		Modified algorithm	
	Global		Local							
N	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
50	53.757	142.049	4.7165	12.359	16.7003	22.8088	0.00525	0.015	0.00225	0.015
100	171.39	437.239	14.97	39.889	299.368	408.999	0.01355	0.031	0.006	0.015
150	595.46	1165.21	32.318	63.789	1874.72	2854.32	0.04075	0.046	0.01505	0.031
200	1690.4	3289.5	32.993	94.308	6085.92	6734.67	0.09325	0.109	0.03625	0.062
250	3118.1	5751.46	25.821	82.988			0.1968	0.343	0.07035	0.093
500							1.7258	1.934	0.5529	0.733
1000							20.7865	21.964	6.011	7.363
1500							86.8852	91.642	25.9719	30.529
2000							262.69	279.77	79.4926	92.476
2500							668.438	710.517	208.057	248.617

Columns 2-3 of Table 5.5 show the average and maximum percentage difference of the objective value of the solution returned by LINGO's local solver from the problem's global optimal solution objective value. The next two columns of the same table show the average (rounded to the nearest integer) and the maximum number of feasible value combinations of variables z_g and z_a. Of course, these results are always the same for both the solution algorithm that we propose and the decoupling LINGO model. We explain the results shown in the last two columns of Table 5.5 in the next subsection.

Table 5.5: Quality of the solutions returned by LINGO's local solver and number of value combinations for variables z_g and z_a

	LINGO Local		Proposed algorithm		Modified algorithm	
	% Obj		Combinations		Combinations	
N	Avg	Max	Avg	Max	Avg	Max
50	116.969	1109.23	42	49	24	32
100	327.442	2043.78	145	169	66	93
150	244.147	1166.53	333	370	134	178
200	473.786	1076.09	587	637	221	315
250	551.545	1476.18	925	990	343	462
500			3561	3825	1123	1438
1000			14712	14858	4265	5175
1500			31866	32905	9410	10861
2000			56905	59059	16914	19149
2500			89221	92057	26676	31084

The superiority of the solution algorithm that we propose becomes immediately clear, since its computational times are significantly lower than those of both LINGO models. As the results of Table

5.4 demonstrate, the computational savings increase considerably for large scale problem instances, for which the application of LINGO appears impracticable. This is partially due to the fact that the increase in the number of feasible value combinations of variables z_g and z_a is quite moderate as the problem size increases, and partially due to the efficiency of Procedure Sweep. The variability of the solution times appears higher in the case of LINGO than in the case of the algorithm that we propose. Additionally, the original LINGO model with the local solver invoked has a rather unusual behavior, since its average computational requirements for $N = 250$ are lower than those for $N = 150$ and $N = 200$. This is an indication that the total computational effort also depends on the specific characteristics of each problem instance besides its size. With the exception of the problems with $N = 50$, the average computational requirements of the decoupling LINGO model are higher than those of the original LINGO model with the global solver invoked. This is an indication that the significant computational savings of our solution algorithm should be attributed more to the efficiency of Procedure Sweep than to the decoupling of the original problem into smaller subproblems alone.

As expected, the computational requirements of LINGO's local solver are significantly lower than those of the global solver. This comes at a price, however, as columns 2 and 3 of Table 5.5 verify, since the objective value of the solution returned by the local solver is on the average approximately between 100 and 550% higher than that of the global optimal solution. In the worst case, this percentage difference increases to approximately 2000%. One way to improve the quality of the solutions returned by LINGO's local solver is to increase the number of "*multistart solver attempts*" on the "*Global Solver*" tab, but this also increases the computational time required for termination. In general, as this number increases, the results returned by the local solver resemble those returned by the global solver in terms of solution quality and computational requirements.

5.4.3 Algorithmic enhancements

Our extensive computational experience with the proposed solution algorithm and LINGO has provided considerable evidence suggesting that the 2-dimensional "*cost-matrix*" with rows the feasible values of variable z_g, columns the feasible values of variable z_a, and elements the optimal total deviation index values for each particular combination of z_g and z_a, may possess a special type of convexity called row and column convexity. More specifically, let $TC(g,a)$ be the problem's optimal total deviation index value for the combination with $z_g = g$ and $z_a = a$. The corresponding matrix is row convex if $TC(g,a) < TC(g,a+1)$ implies $TC(g,a+1) \leq TC(g,a+i)$ for every feasible $i \geq 2$ and if $TC(g,a) < TC(g,a-1)$ implies $TC(g,a-1) \leq TC(g,a-i)$ for every feasible $i \geq 2$. Similarly, the corresponding table is column convex if $TC(g,a) < TC(g+1,a)$ implies $TC(g+1,a) \leq TC(g+i,a)$ for every feasible $i \geq 2$ and if $TC(g,a) < TC(g-1,a)$ implies $TC(g-1,a) \leq TC(g-i,a)$ for every feasible $i \geq 2$.

Despite numerous and tedious attempts, we haven't been able to develop a formal mathematical proof that establishes the validity of this property. On the other hand, despite extensive experimentation, we haven't been able to discover a single counterexample that disproves it either. If this property is indeed valid, then we can exploit it in order to improve the computational performance of the proposed solution algorithm considerably. To show this, we developed a simple modification of this algorithm, which does not compute the optimal total deviation index value for every feasible combination of z_g and z_a; instead, this algorithm computes this value for the middle element of each row (or column using a specific simplistic rule) of the cost-matrix and terminates its search within the same row (or column) as soon as it has established, assuming that the cost-matrix is indeed row and column convex, that no further improvement on the objective can be accomplished in the same row (or column). After this procedure is repeated for all rows (or columns) of the matrix, the best incumbent solution is returned by the algorithm. The last two columns of Table 5.4 present the average and maximum computational times of this modified algorithm, and the last two columns of Table 5.5 present the average and maximum number of value combinations of variables z_g and z_a for which this algorithm computed the total deviation index.

These results show that if the cost-matrix is indeed row and column convex, then the solution algorithm that we propose can be significantly expedited by cleverly incorporating this property into its original design. The modified algorithm returned the global optimal solution in every problem instance out of the 200 on which it was applied, giving us, for one thing, strong evidence that this is indeed true. Given this intuition, we believe that future research should investigate whether this property is actually valid or not. If it turns out to be, a suitable modification of the proposed solution algorithm that exploits this result to the greatest extent should be developed, so that additional computational benefits can be gained. Given the deficiency that stems from the lack of a formal proof for the validity of this property, we did not develop a sophisticated design for the cost-matrix search, which explains the rather simplistic design that we present above.

5.5 Summary

In this chapter, we developed a mixed integer nonlinear model for flight and maintenance planning of a group of aircraft that comprise a unit. The objective is to provide a balanced allocation of the flight load and the maintenance capacity to each individual aircraft, so that the long term availability of the unit is kept at a high and steady level. The formulation that we propose is based on a suitable modification of an existing graphical heuristic tool for addressing this problem. Utilizing the problem's special structure, we also developed an exact search algorithm for its solution. Our computational results demonstrate the superiority of the proposed algorithm over a commercial optimization package.

Chapter 6 Single Objective Multi-Period FMP Problem

6.1 Introduction

In the previous chapter, we developed an exact solution algorithm for the single period FMP problem. Motivated by the fact that the flight load requirements are typically issued over a planning horizon of 6 monthly periods, in this chapter we develop an exact solution algorithm for the muti-period version of this problem. This algorithm is capable of identifying the optimal solution of considerably large realistic problems in very reasonable computational times.

Initially, the algorithm obtains a valid upper bound on the optimal fleet availability by solving a simplified relaxation of the original problem. In subsequent iterations, this bound is gradually reduced, until a feasible solution is identified. Solutions encountered along the search procedure, which cannot be optimal because they are infeasible, are excluded from further consideration through the addition of suitable valid inequalities (cuts). The algorithm terminates when the first feasible solution that attains the current fleet availability bound is identified, which, naturally, comprises the optimal solution of the problem.

The remainder of the chapter is structured as follows. In Section 6.2 we present the mixed integer linear programming (MILP) formulation for the multi-period version of the FMP problem. In Section 6.3 we develop the proposed solution algorithm, while in Section 6.4 we present experimental results evaluating its computational performance. In Section 6.5 we discuss some interesting model extensions, and finally, in Section 6.6 we summarize this chapter.

6.2 Single objective multi-period FMP Model (S-FMPh)

The mixed integer linear programming model for the formulation of the multi-period FMP problem utilizes the following mathematical notation:

Sets:

N: set of aircraft in the unit, indexed by n.

Parameters:

T : length of the planning horizon,

S_t : flight load requirements in time period t,

B_t : time capacity of the maintenance facility in time period t,

C : space capacity of the maintenance facility,

Y : residual flight time of an aircraft immediately after it exits the maintenance facility,

G : residual maintenance time of an aircraft immediately after it enters the maintenance facility,

$A1_n$: state (0/1) of aircraft n at the beginning of the planning horizon,

$Y1_n$: residual flight time of aircraft n at the beginning of the planning horizon,

$G1_n$: residual maintenance time of aircraft n at the beginning of the planning horizon,

X_{max} : maximum flight time of an aircraft in a single time period,

Y_{min} : lower bound on the residual flight time of an available aircraft,

G_{min} : lower bound on the residual maintenance time of a grounded aircraft,

K : a sufficiently large number.

Decision Variables:

$a_{n,t}$: binary decision variable equal to 1 if aircraft n is available in time period t, and 0 otherwise,

$y_{n,t}$: residual flight time of aircraft n at the beginning of time period t,

$x_{n,t}$: flight time of aircraft n in time period t,

$g_{n,t}$: residual maintenance time of aircraft n at the beginning of time period t,

$h_{n,t}$: maintenance time of aircraft n in time period t,

$d_{n,t}$: binary decision variable equal to 1 if aircraft n exits the maintenance facility at the beginning of time period t, and 0 otherwise,

$f_{n,t}$: binary decision variable equal to 1 if aircraft n enters the maintenance facility at the beginning of time period t, and 0 otherwise,

q_t, $p_{n,t}$, $r_{n,t}$: auxiliary binary decision variables.

Utilizing the above notation, the proposed FMP model is formulated as follows:

Problem S-FMP$_h$:

$$\text{Max } CFA_h = \sum_{t=2}^{T+1} \sum_{n=1}^{|N|} y_{n,t} \quad (6.2.1)$$

$$\text{s.t. } y_{n,t+1} = y_{n,t} - x_{n,t} + Yd_{n,t+1},\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.2)$$

$$d_{n,t+1} \geq a_{n,t+1} - a_{n,t},\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.3)$$

$$a_{n,t+1} - a_{n,t} + 1.1(1 - d_{n,t+1}) \geq 0.1,\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.4)$$

$$g_{n,t+1} = g_{n,t} - h_{n,t} + Gf_{n,t+1},\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.5)$$

$$f_{n,t+1} \geq a_{n,t} - a_{n,t+1},\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.6)$$

$$a_{n,t} - a_{n,t+1} + 1.1(1 - f_{n,t+1}) \geq 0.1,\ n = 1,\ldots,|N|,\ t = 1,..,T \quad (6.2.7)$$

$$\sum_{n=1}^{|N|} x_{n,t} = S_t,\ t = 1,\ldots,T \quad (6.2.8)$$

$$\sum_{n=1}^{|N|} h_{n,t} \le B_t,\ t = 1,...,T \tag{6.2.9}$$

$$\sum_{n=1}^{|N|} (1 - a_{n,t}) \le C,\ t = 2,...,T+1 \tag{6.2.10}$$

$$B_t \le \sum_{n=1}^{|N|} h_{n,t} + K(1 - q_t),\ t = 1,...,T \tag{6.2.11}$$

$$\sum_{n=1}^{|N|} g_{n,t} \le \sum_{n=1}^{|N|} h_{n,t} + Kq_t,\ t = 1,...,T \tag{6.2.12}$$

$$y_{n,t} + Kp_{n,t} \le K,\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.13}$$

$$a_{n,t+1} \le (y_{n,t} - x_{n,t})K + Kp_{n,t},\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.14}$$

$$g_{nt} + Kr_{n,t} \le K,\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.15}$$

$$1 - a_{n,t+1} \le (g_{n,t} - h_{n,t})K + Kr_{n,t},\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.16}$$

$$y_{n,t} \le Ya_{n,t},\ n = 1,\dots,|N|,\ t = 2,...,T+1 \tag{6.2.17}$$

$$g_{n,t} \le G(1 - a_{n,t}),\ n = 1,\dots,|N|,\ t = 2,...,T+1 \tag{6.2.18}$$

$$x_{n,t} \le X_{max}a_{n,t},\ n = 1,\dots,|N|,\ t = 1,...,T \tag{6.2.19}$$

$$y_{n,t} \ge Y_{min}a_{n,t},\ n = 1,\dots,|N||,\ t = 2,...,T+1 \tag{6.2.20}$$

$$g_{n,t} \ge G_{min}(1 - a_{n,t}),\ n = 1,\dots,|N|,\ t = 2,...,T+1 \tag{6.2.21}$$

$$x_{n,t} \le y_{n,t},\ n = 1,\dots,|N|,\ t = 1,...,T \tag{6.2.22}$$

$$h_{n,t} \le g_{n,t},\ n = 1,\dots,|N|,\ t = 1,...,T \tag{6.2.23}$$

$$a_{n,1} = A1_n,\ n = 1,\dots,|N| \tag{6.2.24}$$

$$y_{n,1} = Y1_n,\ n = 1,\dots,|N| \tag{6.2.25}$$

$$g_{n,1} = G1_n,\ n = 1,\dots,|N| \tag{6.2.26}$$

$$x_{n,t},\ h_{n,t} \ge 0;\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.27}$$

$$y_{n,t},\ g_{n,t} \ge 0;\ n = 1,\dots,|N|,\ t = 2,..,T+1 \tag{6.2.28}$$

$$p_{n,t},\ r_{n,t},\ q_t \text{ binary},\ n = 1,\dots,|N|,\ t = 1,..,T \tag{6.2.29}$$

$$a_{n,t}, d_{n,t}, f_{n,t} \text{ binary}, n = 1,\ldots,|N|, t = 2,..,T+1 \tag{6.2.30}$$

The objective function (6.2.1) is similar to the objective function (4.2.3) of the multi-objective model presented in Section 4.2, while the constraint set is similar to the constraint set of that model. Nevertheless, for reasons of completeness and clarity, we also provide a short description of the above model next. Two key differences that this model exhibits with respect to the model of Section 4.2 stem from the fact that it refers to a single unit not comprised of distinct squadrons, and from the fact that constraint (6.2.8) imposes the exact flight load satisfaction, as opposed to the tolerance allowed on the flight load satisfaction by constraint (4.2.15). Section 6.5 provides a brief discussion on how the multi-squadron case, as well as the case of non-strict flight load satisfaction, can be handled within the context of the present model.

The objective function (6.2.1) maximizes the cumulative residual flight time availability (CFA_h) of the unit, defined as the sum of the individual residual flight time availabilities of all time periods. The availability of the first time period is not included in the objective since it is pre-determined upon the application of the model, whereas that of time period T+1 is included, ensuring a smooth transition into the next planning horizon. For each time period, constraint set (6.2.2) updates the residual flight time of each aircraft. When an aircraft exits the maintenance facility at the beginning of time period t+1, binary variable $d_{n,t+1}$ takes the value 1, while the residual flight time of that aircraft is reset to Y. Parameter Y, also called *phase interval*, represents the total flight time until the next maintenance inspection. Similarly, for each time period, constraint set (6.2.5) updates the residual maintenance time of each aircraft. When an aircraft is grounded for maintenance inspection at the beginning of time period t+1, binary variable $f_{n,t+1}$ takes the value 1, while the residual maintenance time of that aircraft is reset to G. Parameter G represents the total service time of the maintenance inspection.

Constraint sets (6.2.3), (6.2.4), (6.2.6) and (6.2.7) utilize the values of variables $a_{n,t}$ in order to assign proper values to variables $d_{n,t}$ and $f_{n,t}$. More specifically, ($a_{n,t}$, $a_{n,t+1}$) can be (0,1), (0,0), (1,0) or (1,1), making ($a_{n,t+1}$ - $a_{n,t}$) equal to 1, 0, -1 and 0, respectively. When this difference is equal to 1, variable $d_{n,t+1}$ should be equal to 1, which is ensured by constraint set (6.2.3). Otherwise, $d_{n,t+1}$ should take the value 0, which is ensured by constraint set (6.2.4). Similarly, when this difference is equal to -1, variable $f_{n,t+1}$ should be equal to 1, which is ensured by constraint set (6.2.6). Otherwise, $f_{n,t+1}$ should take the value 0, which is ensured by constraint set (6.2.7).

Constraint set (6.2.8) ensures satisfaction of the flight load requirements in each time period, while constraint sets (6.2.9) and (6.2.10) ensure that the restrictions pertaining to the time and space, respectively, capacity of the maintenance facility are respected. For each time period, constraint sets (6.2.11) and (6.2.12) impose the full utilization of the facility's time capacity if it does not suffice for finishing the service of the grounded aircraft, or the full completion of this service if the opposite is true. This is achieved through the utilization of the auxiliary binary variable q_t, which assists in setting the total maintenance time provided by the facility in time period t equal to the minimum between the total service time requirements and the total time capacity of the maintenance facility in that period.

Constraint sets (6.2.13) and (6.2.14) state that if the residual flight time of an available aircraft drops to 0, then this aircraft must be grounded for service. The auxiliary binary variable $p_{n,t}$ becomes equal to 0 if $y_{n,t} > 0$, which forces $a_{n,t+1}$ to 0-value if $y_{n,t} = x_{n,t}$. Similarly, constraint sets (6.2.15) and (6.2.16) state that if the residual maintenance time of a grounded aircraft drops to 0, then this aircraft must exit the facility and become available. The auxiliary binary variable $r_{n,t}$ becomes equal to 0 if $g_{n,t} > 0$, which forces $a_{n,t+1}$ to 1 if $g_{n,t} = h_{n,t}$.

Constraint set (6.2.17) imposes an upper bound equal to Y on the residual flight time of each available aircraft, and sets the residual flight time of each grounded aircraft equal to 0. Similarly, constraint set (6.2.18) imposes an upper bound equal to G on the residual maintenance time of each grounded aircraft, and sets the residual maintenance time of each available aircraft equal to 0. Constraint set (6.2.19) imposes an upper bound equal to X_{max} on the flight time of each available aircraft in a single time period, and sets the flight time of each grounded aircraft equal to 0. This upper bound is usually imposed due to technical restrictions. Constraint set (6.2.20) imposes a lower bound equal to Y_{min} on the residual flight time of each available aircraft, while constraint (6.2.21) imposes a lower bound equal to G_{min} on the residual maintenance time of each grounded aircraft. This way, an aircraft cannot end-up with a positive but negligible residual flight or maintenance time. Constraint set (6.2.22) states that the total flight time of an aircraft in a single time period cannot be larger than its residual flight time at the beginning of this time period. Similarly, constraint set (6.2.23) states that the total maintenance time of an aircraft in a single time period cannot be larger than its residual maintenance time at the beginning of this time period.

Constraint sets (6.2.24), (6.2.25) and (6.2.26) are used to initialize the status of the aircraft at the beginning of the planning horizon. It should be noted that variables $d_{n,1}/f_{n,1}$ are never used, since variables $y_{n,1}/h_{n,1}$ are directly updated to depict the exit/entrance of an aircraft from/into the maintenance facility at the beginning of the planning horizon. Finally, constraints (6.2.27) and (6.2.28) impose the non-negativity of the continuous decision variables, while constraints (6.2.29) and (6.2.30) impose the integrality of the binary decision variables.

Let x be a solution to the problem formulation (6.2.1)-(6.2.30) introduced above, $CFA_h(x)$ be the cumulative residual flight time availability realized by this solution, and X be the set of all feasible solutions. In short, the S-FMP$_h$ problem introduced above can be expressed as:

$$\text{Max } CFA_h(x) \quad \text{(S-FMP}_h\text{)}$$
$$\text{s.t. } x \in X$$

6.3 Solution methodology

The solution algorithm that we develop for the FMP problem utilizes the fact that the CFA_h of the unit depends solely on the combination of aircraft that enter and exit the maintenance station over the planning horizon, and that the number of such combinations is finite. As a consequence, the domain comprised of possible CFA_h values is a discrete set. Initially, the algorithm identifies a valid upper bound on the optimal CFA_h by solving a simplified relaxation of the original problem; then this bound is gradually decreased, until a feasible flight and maintenance plan that attains it is identified.

It can be shown that if Problem (6.2.1)-(6.2.30) has one or more optimal solutions, then at least one of them preserves a steady rotation of aircraft into and out of the maintenance station, in non-decreasing order of their residual flight/maintenance times. In practice, no such restriction is present. Aircraft are allowed to enter and exit the maintenance station in any feasible order, while their indices are updated accordingly to represent their resulting relative order. With this in mind, the index representing the relative order of each aircraft at the beginning of the next period should be a decision variable allowed to take any feasible value. Adding this degree of freedom, however, complicates the solution of the problem unnecessarily, without providing any advantage whatsoever; therefore, the algorithm that we develop next adopts the assumption that a steady rotation of aircraft into and out of the maintenance station is preserved.

It is relatively easy to prove that this assumption does not affect the optimal objective of the problem. In fact, this result has been proven in Proposition 5.1 for the single-period FMP problem studied in Section 5. The proof involves exchanging the actions performed on any two aircraft for which this order is not preserved, so as to reinstate it. This can be done straightforwardly, without altering the optimal objective. Repeating this exchange for all such pairs of aircraft leads to an alternative optimal solution for which this order is preserved. As a result, the validity of the proposition is confirmed.

In general, several distinct aircraft combinations can result in the same CFA_h. Each time one such combination is identified, the algorithm checks whether it is feasible, i.e., whether it can be realized by a feasible flight and maintenance plan. If this check is successful, then the algorithm terminates with the solution determined by this combination being optimal. If not, a suitable cut is added to the model, excluding this combination from further consideration. All the cuts pertaining to the same CFA_h level remain active for as long as this level remains constant. If, at some point, the currently considered CFA_h level is proven infeasible (i.e., if it cannot be attained by any feasible aircraft combination), then the search for the optimal solution continues to the next (lower) level from the CFA_h domain set. This renders the cuts associated with the previous level redundant, which are subsequently suppressed.

To check a particular aircraft combination for feasibility, we utilize the original formulation, after adjusting accordingly the model constraints to force its realization. Alternatively, one could directly utilize Problem (6.2.1)-(6.2.30) in order to check whether a feasible aircraft combination that attains a particular CFA_h level exists. As it turns out, however, identifying the candidate aircraft combination first and then checking its feasibility is more efficient, at least for the realistic problems that we test in this dissertation. This is mainly due to the fact that the model utilized for the initial feasibility check on the identified combination is considerably more simplistic than the original model comprised of (6.2.1)-(6.2.30). As a result, the former model terminates quite fast, whereas the latter one occasionally requires considerable time in order to terminate. Taking also into consideration the fact that the number of combinations that the algorithm encounters in the case of realistic problems is quite small (the related results are presented in Section 6.4), it is not very surprising that this is the case. On the other hand, when this number increases, the opposite behavior is observed, i.e., it is more efficient to apply the original model directly instead. With this discussion in mind, the following three subsections portray in detail each step of the proposed algorithm, while the last one illustrates its application on a small case study.

6.3.1 Bounding the optimal CFA_h

As determined by constraint (6.2.8), the residual flight time availability of the unit reduces by the associated flight load in each time period of the planning horizon, independently of how this flight load is distributed across the aircraft of the unit. Based on this observation, we claim that, as far as the actions of the maintenance station are concerned, the maximum possible CFA_h level is attained when the maintenance crew works continuously on the grounded aircraft with the lowest residual maintenance time until its service is completed. To get more insight into why this is true, note that interrupting the service of a grounded aircraft once this has begun may lead to a sub-optimal solution, since it can delay the addition of this aircraft's phase interval to the fleet availability of the unit. This would clearly result in lower CFA_h, since the number of aircraft exiting the station at any individual time period is more heavily weighted in the objective function than that of any succeeding one. Of course, the service of an aircraft may be spread out over more than one time periods if the station's time capacity is not sufficient.

Let en_t and ex_t be the number of aircraft that enter and exit, respectively, the maintenance station at the beginning of time period t (t = 2,…,T+1), and $cx_t = \sum_{k=2}^{t} ex_k$ be the cumulative number of aircraft that exit the maintenance station from the beginning of the planning horizon up to time period t (t = 2,…,T+1). In order to show that a first-in-first-out maintenance policy that serves continuously the aircraft with the lowest residual maintenance time until its service is completed always leads to the optimal solution of the problem, we prove next the following result:

Proposition 6.1: Maximizing the objective function of Problem (6.2.1)-(6.2.30) is equivalent to maximizing $\sum_{t=2}^{T+1} cx_t$.

Proof. For $t = 2,\ldots,T+1$, the *individual* residual flight time availability of the unit in time period t is equal to $\sum_{n\in N} y_{n,1} - \sum_{k=1}^{t-1} S_k + \sum_{k=2}^{t}(Yex_k)$. Therefore, the objective function value of problem (6.2.1)-(6.2.30) is equal to $\sum_{t=2}^{T+1}(\sum_{n\in N} y_{n,1} - \sum_{k=1}^{t-1} S_k + \sum_{k=2}^{t}(Yex_k)) = T\sum_{n\in N} y_{n,1} - \sum_{t=1}^{T}((T-t+1)S_t) + \sum_{t=1}^{T}((T-t+1)Yex_{t+1})$.

The only non-constant term subject to optimization in this expression is the last summation, which is equal to $TYex_2 + (T-1)Yex_3 + ... + Yex_{T+1} = Y(Tex_2 + (T-1)ex_3 + ... + ex_{T+1})$. Therefore, maximizing the objective function of Problem (6.2.1)-(6.2.30) is equivalent to maximizing the expression $(Tex_2 + (T-1)ex_3 + ... + ex_{T+1}) = \sum_{t=1}^{T}(T-t+1)ex_{t+1}$.

Since $\sum_{t=2}^{T+1} cx_t = \sum_{t=2}^{T+1}(\sum_{k=2}^{t} ex_k) = \sum_{t=1}^{T}(T-t+1)ex_{t+1}$, the validity of the proposition is established.

As a consequence of Proposition 6.1, the following result regarding the optimal policy of the maintenance station is now evident:

Corollary 6.1: No other maintenance policy can result in higher objective value for Problem (6.2.1)-(6.2.30) than a first-in-first-out policy that always services the grounded aircraft with the lowest residual maintenance time continuously until its service is completed.

The validity of the corollary results directly from the fact that no other maintenance policy can result in larger value for $\sum_{t=2}^{T+1} cx_t$. Of course, depending on the particular problem instance, the optimal CFA_h may also be attainable by another maintenance policy; the important finding that we utilize in the remainder of this work, however, is that the optimal solution will never be overlooked if this maintenance policy is adopted. If the aircraft with the lowest residual maintenance time is not unique, the particular aircraft selection can be made arbitrarily.

The total number of aircraft that exit the station cannot be larger than the corresponding number that results when the total number of aircraft that enter the maintenance station is the maximum possible. Therefore, to compute a valid upper bound on the optimal CFA_h, we enforce the maximum possible flow of aircraft into the maintenance station by grounding each available aircraft as early as possible. This ensures that the time capacity of the station will be fully utilized and that the maximum possible number of aircraft maintenance services will be completed as early as possible. To expedite the performance of the algorithm, a small subset of the original model constraints are taken into consideration for the calculation of this bound. This implies that the associated aircraft combination that will be identified will not necessarily be feasible. Of course, a subsequent check that confirms or disproves feasibility is always performed on each such combination.

The following pseudocode outlines the steps of the procedure for obtaining the valid upper bound on the optimal CFA_h, and facilitates the establishment of its validity which is proven in Proposition 6.2. To keep this pseudocode simple and readable, we utilize the following additional mathematical notation:

C_{res} : current number of empty aircraft spots at the maintenance station (residual space capacity),

B_{res} : currently unused time capacity of the maintenance station (residual time capacity),

TS_t : cumulative flight load requirements of time periods $1,\ldots,t$ (by convention, $TS_0 = 0$),

TS_ent_t : cumulative flight load that has been fulfilled by the aircraft that have been grounded in time periods $1,\ldots,t$ (by convention, $TS\_ent_0 = 0$),

$tlast_n$: the most recent time period aircraft n exited the maintenance station (by convention, $tlast_n = 1$ if aircraft n has not exited the maintenance station in the current planning horizon yet),

$ylast_n$: auxiliary variable that is set equal to Y if $tlast_n > 1$, and $y_{n,1}$ otherwise,

flag : auxiliary boolean variable.

Procedure CFA_h-UB

Step 0: Preprocessing

order the available/grounded aircraft in non-decreasing order of their residual flight/maintenance times

set $CFA_h = T\sum_{n\in N} y_{n,1}$; **set** $C_{res} = C - \sum_{n\in N}(1-a_{n,1})$; **set** $TS_0 = 0$; **set** $TS\_ent_0 = 0$;

$\forall n \in N$ **set** $tlast_n = 1$;

Step 1: Iteration

for $t = 1$ to T **do**

set $TS_t = TS_{t-1} + S_t$; **set** $TS\_ent_t = TS\_ent_{t-1}$; **set** $B_{res} = B_t$; **set** $ex_{t+1} = 0$;

  while $B_{res} > 0$ **&** additional grounded aircraft exist **do**

    select the grounded aircraft with the lowest residual maintenance time (if more than one such aircraft exist, select one arbitrarily); let q be the index of this aircraft

    set $g_{q,t+1} = g_{q,t} - \min(g_{q,t},B_{res})$; **set** $B_{res} = B_{res} - \min(g_{q,t},B_{res})$;

    if $g_{q,t+1} = 0$ **then**

      set $ex_{t+1} = ex_{t+1} + 1$; **set** $C_{res} = C_{res} + 1$;

      remove aircraft with index q from the set of grounded aircraft and **add** it to the set of aircraft which are available at the beginning of time period t+1; **set** $y_{q,t+1} = Y$; **set** $tlast_q = t+1$

    end if

end while

set $CFA_h = CFA_h + (T\text{-}t+1)Yex_{t+1}$; **set** $en_{t+1} = 0$; **set** *flag* = true;

while ($C_{res} > 0$) **&** additional available aircraft exist **&** (*flag* = true) **do**

select the available aircraft with the lowest residual flight time (if more than one such aircraft exist, select one arbitrarily); let q be the index of this aircraft

if ($tlast_q > 1$) **then set** $ylast_q = Y$; **else set** $ylast_q = y_{q,1}$; **end if**

if ($ylast_q > X_{max*}(t\text{-}tlast_q+1)$) **or** ($ylast_q + TS\_ent_t > TS_t$) **then**

set *flag* = false;

else

set $TS\_ent_t = TS\_ent_t + ylast_q$; **set** $en_{t+1} = en_{t+1} + 1$; **set** $C_{res} = C_{res} - 1$;

remove aircraft with index q from the set of available aircraft and **add** it to the set of aircraft which are grounded at the beginning of time period t+1; **set** $g_{q,t+1} = G$;

end if

end while

set $CFA_h = CFA_h - (T\text{-}t+1)S_t$;

end for

Procedure *CFA*CFA_h-*UB* performs two main actions in each time period t. First, it computes the number of aircraft that will finish their service and exit the station at the end of time period t. This is straightforward, given the complete knowledge of the aircraft that are grounded at the beginning of time period t, and Corollary 6.1. Next, it examines the available aircraft in non-decreasing order of their residual flight times, and checks which of them can be grounded. The grounding of a particular aircraft is feasible only if the remaining aircraft are sufficient for satisfying the flight load requirements. Variables $tlast_n$ and $ylast_n$ are crucial for this check. If $X_{max} = 50$, an aircraft with residual flight time 80 hours needs at least two time periods in order to enter the maintenance station. Moreover, if $Y = 300$, an aircraft that has just exited the maintenance station needs at least $Y/X_{max} = 300/50 = 6$ time periods in order to be grounded for service again.

Procedure CFA_h-*UB* interrupts the flight time allocation in a particular time period as soon as the first aircraft that cannot be feasibly grounded at the end of this period is identified, since this implies that no other aircraft can be feasibly grounded either. In the above pseudocode, this is signified by variable *flag* which is set equal to false whenever this situation is detected. To ensure that no feasible solution is ever overlooked, this check is performed separately for each time period. The key assumption when this is done in time period t is that the residual flight time of each aircraft n is equal to $ylast_n$ and that no flight time has been allocated yet to this aircraft in time periods $tlast_n,\ldots,t$.

This way, the flight time allocation is limited only to aircraft for which their earliest grounding time period has been determined, which guards against taking decisions that may turn out to be sub-optimal in future time periods. This is exactly why parameters $tlast_n$ and $ylast_n$ are introduced in the first place. By maintaining the values of these two variables for each aircraft n, we are able to take into account every feasible flight time allocation to aircraft n in time periods $tlast_n,\ldots,t$, which ensures that no feasible aircraft combination will be overlooked. This enables Procedure CFA_h-UB to compute the maximum possible number of grounded aircraft that can finish their service in each time period, as well as the maximum possible number of available aircraft that can be grounded for service. The above discussion leads to the following crucial result.

Proposition 6.2: Procedure CFA_h-UB provides a valid upper bound on the optimal objective value (CFA_h) of problem (6.2.1)-(6.2.30).

Proof. As far as the maintenance decisions are involved, Corollary 6.1 ensures that the computed CFA_h will be the maximum possible. For each time period, the above procedure computes an upper bound on the maximum possible number of aircraft that can cumulatively enter the maintenance station from the beginning of the planning horizon up to this period, thus maximizing the flow of aircraft into the maintenance station. This implies that the actual number of aircraft that enter the maintenance station for service from the beginning of the planning horizon up to the end of time period t for $t = 2,\ldots,T+1$ in the optimal solution of the problem cannot be larger than the corresponding number that results from Procedure CFA_h-UB. Therefore, the optimal CFA_h cannot be larger than the one provided by Procedure CFA_h-UB.

Besides establishing a valid upper bound on the optimal CFA_h, Procedure CFA_h-UB also identifies a particular combination of aircraft that enter and exit the maintenance station in each time period of the planning horizon. In what follows, we call this the *nominal* combination, independently of whether it is feasible or not. Naturally, the nominal is the first combination that the algorithm checks for full feasibility. The particular details of this procedure are presented in the next subsection.

6.3.2 Checking a particular aircraft combination for feasibility

Checking a particular aircraft combination for feasibility is trivial. Keeping in mind that the order of aircraft is preserved, the check of whether a particular aircraft combination is feasible reduces to a check of whether there exists a feasible flight and maintenance plan that realizes this combination. This is equivalent to checking the original problem formulation for feasibility after fixing the values of all the discrete decision variables which are determined by this combination, i.e., decision variables $a_{n,t}$, $d_{n,t}$, $f_{n,t}$, $p_{n,t}$, $r_{n,t}$, and q_t. This simplifies things considerably, since it eliminates completely the combinatorial nature of the original problem; as a result, the feasibility check is not time-consuming.

6.3.3 Adding a cut for the exclusion of a particular aircraft combination

Suppose now that a particular aircraft combination is proven infeasible. In order to investigate whether the currently considered CFA_h level can be attained by a different aircraft combination, we need to add a valid inequality excluding this combination from further consideration. We augment variables en_t and ex_t with a second index that takes the value 0, 1 or 2, as explained next. The index 0 pertains to the nominal combination, the index 1 pertains to the combination at hand that we want to exclude, while the index 2 pertains to any other aircraft combination yet to be discovered by the algorithm. A suitable cut that excludes the infeasible aircraft combination at hand from further consideration is the following: $\sum_{t=2}^{T+1} | en_{t,1} - en_{t,2} | + \sum_{t=2}^{T+1} | ex_{t,1} - ex_{t,2} | \geq 1$. In this expression, $en_{t,2}$ and $ex_{t,2}$ (t = 2,…,T+1) are decision variables, whereas $en_{t,1}$ and $ex_{t,1}$ (t = 2,…,T+1) are parameters with known values. The constraint ensures that the number of aircraft exiting and entering the maintenance station in the next combination that will be discovered by the algorithm will differ from the one at hand in at least one time period. Letting W denote the CFA_h level that is currently being considered, $na_{t,i}$ be the number of grounded aircraft in time period t (t = 2,…,T+1) in the combination that is being excluded (i = 1) and in the next combination to be discovered (i = 2), the following mixed integer formulation can be employed to impose the above cut and provide the next candidate aircraft combination:

$$\sum_{t=1}^{T}((T-t+1)ex_{t+1,2}) = \frac{W - T\sum_{n\in N} y_{n,1} + \sum_{t=1}^{T}((T-t+1)S_t)}{Y} \tag{6.3.1}$$

$$\sum_{k=2}^{t} ex_{k,2} \leq \sum_{k=2}^{t} ex_{k,0},\ t = 2,..,T+1 \tag{6.3.2}$$

$$\sum_{k=2}^{t} en_{k,2} \leq \sum_{k=2}^{t} en_{k,0},\ t = 2,..,T+1 \tag{6.3.3}$$

$$na_{t,2} = na_{t-1,2} + en_{t,2} - ex_{t,2},\ t = 2,..,T+1 \tag{6.3.4}$$

$$na_{t,2} \leq C,\ t = 2,..,T+1 \tag{6.3.5}$$

$$ex_{t,2} \leq na_{t-1,2},\ t = 2,..,T+1 \tag{6.3.6}$$

$$\sum_{t=2}^{T+1} | en_{t,1} - en_{t,2} | + \sum_{t=2}^{T+1} | ex_{t,1} - ex_{t,2} | \geq 1, \tag{6.3.7}$$

$$en_{t,2},\ ex_{t,2},\ na_{t,2} \text{ integer} \geq 0,\ t = 2,..,T+1 \tag{6.3.8}$$

Constraint (6.3.1) fixes the currently considered CFA_h level. Constraint sets (6.3.2) and (6.3.3) impose upper bounds on the cumulative number of aircraft that exit and enter the station, respectively, based on the nominal aircraft combination. Constraint set (6.3.4) updates the number of grounded aircraft based on the number of aircraft that enter and exit the maintenance station. Constraint set (6.3.5) ensures that the space capacity of the maintenance station will not be violated in any time period. Constraint set (6.3.6) states that the number of aircraft exiting the station at the beginning of each time period cannot exceed the number of aircraft that were grounded during the previous time period. Constraint (6.3.7) is the valid cut that excludes the infeasible combination at hand. Of course, if more than one such combinations have been identified, one such cut needs to be added for each of them. Finally, the last constraint set imposes the non-negativity and the integrality of the decision variables.

Consider a particular pair ($en_{t,2}$, $en_{t,1}$) in the above formulation. The following set of constraints, in which z and u are two auxiliary decision variables and K is a sufficiently large number, eliminates the nonlinearities introduced by the corresponding absolute term in constraint (6.3.7):

$$en_{t,2} - en_{t,1} \leq z \leq en_{t,2} - en_{t,1} + Ku \tag{6.3.9}$$

$$en_{t,1} - en_{t,2} \leq z \leq en_{t,1} - en_{t,2} + K(1-u) \tag{6.3.10}$$

$$u \text{ binary} \tag{6.3.10}$$

Essentially, this set of constraints sets z equal to $|en_{t,1} - en_{t,2}|$. If $en_{t,2} > en_{t,1}$ then u takes the value 0 and this is determined by constraint (6.3.9), while constraint (6.3.10) becomes redundant. If $en_{t,2} < en_{t,1}$ then u takes the value 1 and this is determined by constraint (6.3.10), while constraint (6.3.9) becomes redundant. Of course, both values will work for u if $en_{t,2} = en_{t,1}$. Appending one such constraint set for each pair ($en_{t,2}$, $en_{t,1}$) and ($ex_{t,2}$, $ex_{t,1}$), together with a constraint that sets the sum of auxiliary variables z at least equal to 1 is equivalent to constraint (6.3.7), while also preserving the linearity of the formulation.

If the set of constraints (6.3.1)-(6.3.8) augmented with all the cuts that are currently active is proven infeasible, this is an indication that the currently considered CFA_h level cannot be attained by any feasible aircraft combination. In order to compute the next lower CFA_h level that is candidate for optimality, we utilize the same formulation after substituting (6.3.1) with an objective that maximizes the expression $\sum_{t=2}^{T+1}((T-t+1)ex_{t,2})$ subject to an upper bound of $\frac{W - T\sum_{n\in N} y_{n,1} + \sum_{t=1}^{T}((T-t+1)S_t)}{Y}$. The next candidate CFA_h level is discovered this way, which is fixed using constraint (6.3.1), as before. At the same time, all the previous cuts are suppressed because they are rendered redundant.

6.3.4 A small case study

In this section, we illustrate the application of the proposed algorithm on a small case study. Consider a unit comprising of six aircraft, five of which are available and one of which is grounded at the beginning of the planning horizon. Table 6.1 presents the residual flight/maintenance times of the aircraft at the beginning of the six-period planning horizon, while Table 6.2 presents the flight load requirements and the time capacity of the maintenance station in each time period. In Table 6.1, bold-style entries denote maintenance times of grounded aircraft and plain-style entries denote flight times of available aircraft. The values of the other problem parameters are G = 320 hours, Y = 300 hours, C = 3, X_{max} = 50 hours, Y_{min} = 0.1 hours and G_{min} = 0.1 hours.

Table 6.1: Residual flight/**maintenance** times ($y_{n,1}$/$\mathbf{g_{n,1}}$) (hours)

$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$\mathbf{n = 6}$
5	38	186	213	257	**70**

Table 6.2: Flight load requirements and time capacity of the maintenance station

t	1	2	3	4	5	6
S_t	97	115	99	121	121	113
B_t	129	148	154	144	126	135

The valid upper bound on the optimal CFA_h obtained by Procedure CFA_h-UB is equal to 4923. The associated (nominal) aircraft combination is shown in Table 6.3.

Table 6.3: Nominal aircraft combination

t	2	3	4	5	6	7
en_t	2	0	0	1	1	1
ex_t	1	0	0	1	0	1

Next, we check where a feasible solution that realizes this combination exists. To this end, we fix in the original formulation the values of all the decision variables which are determined by this combination, keeping in mind that the order of aircraft into and out of the maintenance station is preserved. These are the variables $a_{n,t}$, $d_{n,t}$ and $f_{n,t}$ for n = 1,…,6 and t = 2,…,7, $p_{n,t}$ and $r_{n,t}$ for n = 1,…,6 and t = 1,…,6, as well as several of the variables $x_{n,t}$, $y_{n,t}$, $g_{n,t}$, $h_{n,t}$ (for example, $x_{1,1}$ = 5, $x_{2,1}$ = 38, $g_{6,1}$ = 70, etc.). The user may choose to add only a proper subset of the constraints enforced by the aircraft combination and let the optimization solver deduce the remaining ones, or opt for a tighter formulation by explicitly adding all the implied constraints. For our small example, the inclusion of the above constraints makes the problem infeasible, because the flight load constraints of time periods 5 and 6 are violated. Therefore, we add a valid-cut that excludes this combination, and we utilize the following set of constraints in order to check if the currently considered CFA_h level (4923) can be attained by another aircraft combination.

$$\sum_{t=1}^{6}((7-t)ex_{t+1}) = 10$$

$$\sum_{k=2}^{t} ex_k \leq 1,\ t = 2,\ 3,\ 4$$

$$\sum_{k=2}^{t} ex_k \leq 2,\ t = 5,\ 6$$

$$\sum_{k=2}^{7} ex_k \leq 3$$

$$na_t = na_{t-1} + en_t - ex_t,\ t = 2,..,7$$

$$na_t \leq 3,\ t = 2,..,7$$

$$ex_t \leq na_{t-1},\ t = 2,..,7$$

$$|en_2 - 2| + |en_3 - 0| + |en_4 - 0| + |en_5 - 1| + |en_6 - 1| + |en_7 - 1| + \\ |ex_2 - 1| + |ex_3 - 0| + |ex_4 - 0| + |ex_5 - 1| + |ex_6 - 0| + |ex_7 - 1| \geq 1$$

$$en_t,\ ex_t,\ na_t \text{ integer} \geq 0,\ t = 2,..,7$$

The aircraft combination shown in Table 6.4 is identified next, for which the value of CFA_h is equal to 4923, as before.

Table 6.4: Second aircraft combination

t	2	3	4	5	6	7
en_t	1	0	0	0	1	1
ex_t	1	0	0	1	0	1

Next, we check whether this aircraft combination can be realized by a feasible flight/maintenance plan. Since it cannot, a new cut is added, excluding it from further consideration. The algorithm continues similarly until the first feasible aircraft combination is identified. This is true for the 12th combination identified, which is the one shown in Table 6.5. This is the optimal combination, while the optimal CFA_h is equal to 4923, i.e. equal to the valid upper bound provided by Procedure CFA_h-*UB*. The complete optimal solution of the problem is found easily by forcing the realization of this combination in the original model formulation.

Table 6.5: Optimal aircraft combination

t	2	3	4	5	6	7
en_t	1	0	1	0	0	1
ex_t	1	0	0	1	0	1

6.4 Computational implementation

In this section, we analyze the computational complexity of the proposed solution algorithm, and we present computational results demonstrating its efficiency. We also compare its performance against that of two popular commercial optimization software packages that can be utilized alternatively for the solution of the problem under consideration. In order to portray the applicability of the proposed solution algorithm and highlight the benefits that can emerge from its practical application, we test the proposed algorithm both on realistic problem instances drawn from the operation of a typical aircraft unit of the HAF, as well as on large scale random problem instances whose size and parameter values differ significantly from those of typical problems arising in practice.

6.4.1 Computational complexity

The computational effort of the proposed solution algorithm comprises of the computational effort required for the calculation of the valid upper bound on the optimal CFA_h, of the computational effort required for testing the feasibility of the aircraft combinations that are encountered, and of the computational effort required for the addition of the necessary valid inequalities. Let A and NA be the total number of aircraft that are initially available and grounded, respectively, and $|N| = A + NA$ be the total number of aircraft. Regarding the computational effort for the calculation of the valid upper bound on the optimal CFA_h, the following result is true:

Proposition 6.3. The computational effort of Procedure CFA_h-UB is $O(A \log(A)) + O(NA \log(NA)) + O(T|N|)$.

Proof. The total time required to arrange the available aircraft in non-decreasing order of their residual flight times and the grounded aircraft in non-decreasing order of their residual maintenance times is $O(A \log(A))$ and $O(NA \log(NA))$, respectively. The total time required to initialize the values of variables $tlast_n$ is $O(|N|)$. The preprocessing phase of Procedure CFA_h-UB performs these actions only once. The total time required to compute the maximum cumulative number of aircraft that can enter and exit the maintenance station for each time period is $O(|N|)$. This action is performed once for each time period of the planning horizon. Therefore, the total computational effort of Procedure CFA_h-UB is $O(A \log(A)) + O(NA \log(NA)) + O(|N|) + O(T|N|) = O(A \log(A)) + O(NA \log(NA)) + O(T|N|)$.

Regarding the computational effort required to test the feasibility of a single aircraft combination, with the values of decision variables $a_{n,t}$, $d_{n,t}$, $f_{n,t}$, $p_{n,t}$, $r_{n,t}$, and q_t known, the problem defined by (6.2.1)-(6.2.30) reduces to finding a feasible solution to a system of linear constraints with continuous decision variables. Therefore, its computational complexity is polynomial in the values of parameters $|N|$ and T that define the size of the problem. On the other hand, the total number of aircraft combinations that must be checked in the worst case is exponential in the values of parameters $|N|$, C, and T. For realistic problems, this is hardly an issue, since the value of T is rather small and remains constant, while C is relatively small as a percentage of $|N|$ (~10%), making the number of aircraft combinations that the algorithm encounters relatively small. In the next section, we discuss how this exponential behavior can be properly handled should it arise.

6.4.2 Computational results

The proposed solution algorithm was implemented in C and its performance was compared against that of CPLEX 12.5 (2012) and LINGO 13.0 (2011), two popular commercial optimization software packages. Our computational experiments were performed on an i5-3330 @ 3.0 GHz Intel processor with 16 GB system memory. Typical wing configurations of the HAF comprise of 60-80 aircraft, a number that can increase up to 100 aircraft in special cases. Neither CPLEX nor LINGO can handle problems of this size in reasonable time; hence we chose 5 smaller values (i.e., 10, 15, 20, 25 and 30) for the total number of aircraft that comprise the unit. On the other hand, the proposed solution algorithm is capable of handling considerable larger problems; therefore, we also tested its performance on more challenging problems with $|N|$ = 50, 100 and 200. The planning horizon was always set equal to six monthly periods, since the flight load of a typical combat unit of the HAF is typically issued over a six-month period. For each of these sizes, we solved 30 random problem instances. Both optimization packages were invoked with default options.

The random generator was specially designed so as to make the generated problems as similar as possible to the realistic ones. The specifics are as follows: although C is equal to approximately $0.1|N|$ in practice (for a group of 60-80 aircraft, the maintenance hangar can typically accommodate 6-8 aircraft), we set it equal to $0.15|N|$ rounded to the nearest integer in order to make the generated problem instances more challenging. This is because our computational experience and the complexity analysis of the previous subsection suggest that, in general, the difficulty of solving a particular problem increases as the value of C increases. The number of initially grounded aircraft, NA, was generated randomly, using a discrete probability function that considered integer values between 0 and C, inclusive. This distribution was negatively skewed, so that larger candidate NA values were assigned higher probabilities. In particular, for $x = 0,\ldots,C$, the probability that the number of grounded aircraft at the beginning of the planning horizon was equal to x was set equal to $(x+1)/(\sum_{x=0}^{C}(x+1))$.

Of course, the number of initially available aircraft, A, was always set equal to $|N|$-NA. The residual flight time of each available aircraft at the beginning of the planning horizon was a random number distributed uniformly in the interval $[Y_{min},Y]$, while the residual maintenance time of each grounded aircraft at the beginning of the planning horizon was a random number distributed uniformly in the interval $[G_{min},G]$.

For each time period t, S_t was uniformly distributed in the interval $[10|N|, 15|N|]$, and B_t was uniformly distributed in the interval $[15|N|, 20|N|]$. This design approximates fairly close the characteristics of realistic problems; the detailed reasons are strictly confidential. Actual values drawn from the real application were used for the other problem parameters, i.e., $Y = 300$, $G = 320$, $X_{max} = 50$, $Y_{min} = 0.1$ and $G_{min} = 0.1$.

Table 6.6 presents the average and maximum computational times of the two commercial software packages and our proposed solution algorithm for these realistic problems. More specifically, columns 2 and 3 of this table show the computational times of LINGO, columns 6 and 7 show the computational times of CPLEX with the single-thread option selected, while columns 8 and 9 show the computational times of CPLEX with the multi-thread option selected (this is the default option). Additionally, columns 10 and 11 show the computational times of the proposed solution algorithm when the original formulation (6.2.1)-(6.2.30) is directly utilized for checking whether a feasible aircraft combination that attains a particular CFA_h level exists, while columns 12 and 13 show the computational times of this algorithm when this check is performed in two steps (according to the procedure described in Section 6.3), instead. The results of columns 4 and 5 are explained in the following paragraphs. When the multi-thread option is selected, CPLEX uses all the available threads (4 in the case of the computer that we used for our experiments). The table is incomplete because the commercial solvers are not able to accommodate all the problem sizes within the 8-hour limit that we enforced.

Table 6.6: Computational times (seconds) comparison for realistic problems

	LINGO		**LINGO with cuts**		**CPLEX single-thread**		**CPLEX multi-thread**		**Proposed algorithm (1-step feas. check)**		**Proposed Algorithm (2-step feas. check)**	
|*N*|	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**
10	0.44	1.73	0.19	0.49	0.05	0.09	0.05	0.09	0.12	0.18	0.10	0.11
15	30.03	174.24	0.40	1.32	1.34	6.89	0.54	2.37	0.15	0.23	0.10	0.11
20			2.43	10.48	1189.57	20550.01	118.72	1321.3	0.47	4.38	0.11	0.13
25			1.65	18.20			1130.14	7794.17	0.30	2.33	0.12	0.14
30			4.26	34.32					0.82	16.46	0.13	0.14
50			4.46	13.15					0.45	0.59	0.18	0.18
100			10.58	18.18					4.48	68.84	0.28	0.29
200									2.73	4.09	0.49	0.52

The superiority of the proposed solution algorithm becomes immediately clear, since its computational times are significantly lower than those of both LINGO and CPLEX. As the results of this table demonstrate, the computational savings increase considerably for large scale problem instances, for which the application of the two software packages appears impracticable. The variability of the computational times appears significant for both CPLEX and LINGO, whereas in the case of our algorithm it appears insignificant. An interesting observation is that, naturally, the performance of CPLEX improves considerably when the multi-thread option is selected instead of the single-thread option. This option should always be used with caution, however, since increasing the number of threads being used increases the probability that the computer will run out of memory on a particular problem that may be otherwise solvable when the single-thread option is selected. As far as the two alternative designs for the proposed solution algorithm are concerned, the last four columns demonstrate that, for these particular problems, the performance of the two-step procedure for identifying feasible aircraft combinations appears superior to that of the direct feasibility check that utilizes the original formulation (6.2.1)-(6.2.30).

While experimenting with alternative solution options, we noticed that the performance of LINGO can improve significantly by introducing additional constraints that impose valid upper bounds on the cumulative number of aircraft that enter and exit the maintenance station from the beginning of the planning horizon up to time period t for $t = 2,\ldots,T+1$. Such bounds can be provided by Procedure CFA_h-UB. Columns 4 and 5 of Table 6.6 show the computational times of LINGO when these bounds are applied. It becomes evident from these results that the improvement is significant, allowing LINGO to handle considerably larger problems within the predefined time limit. In practice, of course, it does not make much sense to adopt this procedure by applying partially the proposed algorithm in order to find these bounds, since its full application exhibits even better performance for all the realistic problems on which it was tested. On the other hand, a similar behavior was not detected for CPLEX (probably due to different strategies followed by the underlying algorithmic procedures), and therefore no similar results are reported.

The high efficiency of the proposed solution algorithm can be partially attributed to the fact that the number of aircraft combinations it encounters is rather small. In turn, this implies that the upper bound provided by Procedure CFA_h-UB is tight. In view of this observation, and in an attempt to test the limits of this algorithm, we tried to generate problem instances for which this bound is considerably looser. In particular, since the algorithm starts by grounding each available aircraft as early as possible, we generated special problems for which although it is possible to ground several aircraft during the first time periods of the planning horizon, the optimal decision is to hold them back and ground them after several subsequent time periods. Tables 6.7 and 6.8 show the data of one problem with such characteristics, for which $A = 10$, $NA = 0$, $T = 6$, $G = 320$ hours, $Y = 300$ hours, $C = 4$, $X_{max} = 50$ hours, $Y_{min} = 0.1$ hours and $G_{min} = 0.1$ hours. The time capacity of the maintenance station is purposely selected to be larger in the first three time periods so as to enable a large number of service completions during these periods, and smaller in the last three time periods, so as to cause delays in the service completions of those periods.

The application of Procedure CFA_h-UB for this problem results in one aircraft being grounded during the first two time periods of the planning horizon and provides an upper bound equal to 7210 on the optimal CFA_h. However, the associated aircraft combination, shown in Table 6.9, is infeasible. In fact, a CFA_h equal to 7210 cannot be realized by any feasible combination; in order to obtain a feasible solution to the problem, the first aircraft grounding must not take place before time period 4.

Table 6.7: Residual flight times ($y_{n,t}$) (hours)

$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
5	250	250	250	250	250	250	250	250	250

Table 6.8: Flight load requirements and time capacity of the maintenance station

t	1	2	3	4	5	6
S_t	450	450	455	200	50	50
B_t	200	200	640	100	100	100

Table 6.9: Nominal aircraft combination

t	2	3	4	5	6	7
en_t	1	0	0	0	4	0
ex_t	0	0	1	0	0	0

Starting from this one, the proposed algorithm identifies in total 165 infeasible aircraft combinations before reaching the optimal. The number of these combinations for each associated CFA_h level is shown in Table 6.10, while the optimal combination that corresponds to a CFA_h level of 6010 is shown in Table 6.11.

Table 6.10: Number of encountered aircraft combinations

CFA level	7210	6910	6610	6310	6010	**Total**
# of A/C comb.	30	45	45	42	3	**165**

Table 6.11: Optimal aircraft combination

t	2	3	4	5	6	7
en_t	0	0	1	0	1	0
ex_t	0	0	0	0	0	0

Looking more closely at the data of this problem, note that if the aircraft with index 1 is grounded at the end of the first period, it cannot exit the maintenance station earlier than the beginning of time period 4. Although the remaining nine aircraft are sufficient for satisfying the flight load requirements of time period 2, all ten aircraft are needed in order to satisfy the flight load requirements of time period 3 (455 hours), due to the upper bound (50 hours) on the flight time of each individual aircraft in a single time period. Therefore, the optimal and only feasible decision is to hold back the first aircraft and ground it at the end of time period 3. The algorithm cannot foresee this, and goes on to examine all the aircraft combinations that can possible arise by grounding this aircraft earlier, before identifying the optimal one. Since the number of these combinations is quite large, this has an adverse effect on the algorithm's performance, which deteriorates considerably.

It should be emphasized that this numerical example violates many of the characteristics that realistic problems exhibit; it is specially designed so as to trouble the proposed solution algorithm as much as possible, and is presented as a reference basis in order to illustrate a case in which the performance of this algorithm is unsatisfactory, as well as the characteristics of a problem for which this situation can arise. The main such characteristic is that there exist one or more aircraft which can be feasibly grounded during the first time periods of the planning horizon, but the optimal decision is to hold them back and ground them after several subsequent time periods.

The fact that the performance of the proposed solution algorithm deteriorates when applied on these non-standard problem instances is confirmed by the results of Table 6.12. In this table, we present the computational times that this algorithm needs in order to find the optimal solution of problems exhibiting the above characteristic, for various problem sizes (30 problem instances were solved for each such size). The problem instances are not completely random, in that several of their parameter values were selected purposely, so as to ensure the existence of this characteristic. Once this had been ensured, however, the remaining parameters were chosen randomly. In particular, the residual flight times of the aircraft, as well as the flight load and the time capacity of the maintenance station in each time period were selected randomly in similar ranges as those of the problem instances tested in Table 6.6, under the additional requirement that the optimal CFA_h level should not be attainable if the first one or two aircraft were grounded before the 4^{th}-5^{th} time period. A trivial trial and error technique was employed in order to achieve that, which simply rejected those random instances for which this was not true. The columns of Table 6.12 are mainly the same with those of Table 6.6, the only difference being that no results are reported regarding the application of LINGO after the incorporation of special cuts, since there is no significant advantage from adding such cuts in the case of these non-standard problems.

Table 6.12: Computational times (seconds) comparison for non-standard problems

	LINGO		**CPLEX single-thread**		**CPLEX multi-thread**		**Proposed algorithm (1-step feas. check)**		**Proposed algorithm (2-step feas. check)**	
$\|N\|$	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**
10	0.04	0.04	0.02	0.03	0.02	0.03	0.04	0.04	13.87	27.64
15	0.05	0.06	0.02	0.03	0.02	0.03	0.05	0.06	26.49	31.71
20	0.06	0.07	0.02	0.03	0.02	0.03	0.07	0.07	27.42	30.42
25	0.08	0.09	0.02	0.03	0.03	0.03	0.08	0.08	29.60	35.24
30	0.09	0.10	0.02	0.03	0.03	0.03	0.09	0.10	29.80	33.24
50	0.15	0.16	0.05	0.06	0.04	0.05	0.15	0.15	33.87	36.40
100	0.30	0.31	0.07	0.08	0.07	0.08	0.29	0.31	47.67	50.44
200	0.64	0.65	0.13	0.14	0.13	0.16	0.61	0.61	64.58	90.50

Besides confirming that the performance of the proposed solution algorithm deteriorates considerably for the non-standard problems, the results of Table 6.12 also show that the opposite is true for CPLEX and LINGO. This observation is important because it suggests that none of the solution algorithms considered in this work can handle efficiently every possible problem of reasonable size that can arise. Additionally, the above results show that, in the case of the non-standard problems, it is preferable to employ directly the original formulation comprised of (6.2.1)-(6.2.30) for identifying feasible aircraft combinations that attain a particular CFA_h level, instead of the two-step procedure which appears superior in the case of the realistic problems. As these results additionally demonstrate, however, the employment of this technique is not even necessary, due to the fact that the performance of CPLEX and LINGO appears superior to that of the proposed solution algorithm, in contrary to what happens in the case of problems with realistic characteristics. Our computational experience suggests that as the actual number of aircraft combinations that the proposed solution algorithm must visit in order to reach the optimal solution increases, the performance of both LINGO and CPLEX improves, whereas that of the proposed solution algorithm deteriorates, and vice versa. In particular, note that, exhibiting superior performance to that of LINGO, CPLEX appears as the most efficient algorithmic solution tool for handling these non-standard problems.

To summarize, for problems with realistic characteristics in which the number of encountered aircraft combinations is quite small, the performance of the proposed algorithm appears superior, whereas for problems with characteristics such as those described above in which this number is considerably larger, the performance of CPLEX/LINGO appears superior. The important conclusion of this analysis is that the proposed algorithm, complemented by generic optimization software such as CPLEX and LINGO, can handle effectively a large variety of FMP problem instances.

6.5 Problem extensions

In this section, we discuss the applicability of the proposed algorithm on some interesting problem extensions, which have been briefly discussed in Section 3.5. In particular, we relax the assumption on the strict satisfaction of the flight load requirements, we extend the single-squadron case to the multi-squadron one, and we discuss alternative problem objectives.

6.5.1 Relaxing the assumption of strict flight load satisfaction

The actual problem definition often calls for satisfaction of the flight load requirements within some predefined tolerance, instead of the strict satisfaction imposed by constraint (6.2.8). When this is the case, constraint set (6.2.8) is expressed as follows instead:

$$LS_t \le \sum_{n\in N} x_{n,t} \le US_t,\ t = 1,...,T\ ,$$

where L and U are two parameters defining the interval in which the actual total flight time of time period t must lie. For example, when L = 0.95 and U = 1.05, a maximum deviation of 5% from the target value of the flight load is allowed in each time period of the planning horizon. The main effect of this modification is that it makes the total flight time of each time period a decision variable instead of a known parameter. Thus, Proposition 6.1 is no longer valid, and maximizing the CFA_h becomes equivalent to maximizing the expression $\sum_{t=2}^{T+1}(cx_t) - \sum_{t=1}^{T}((T-t+1)AS_t)$, where $AS_t = \sum_{n\in N} x_{n,t}$ is the actual flight time in period t.

This implies that the upper bound on the optimal CFA_h obtained by Procedure CFA_h-UB is no longer valid. To compute a valid upper bound on the optimal CFA_h in this case, we can use the value US_t as the total flight time of period t for computing the maximum number of aircraft that can enter and exit the maintenance station over the planning horizon, and the value LS_t as the total flight time of period t for performing the remaining calculations. The validity of the new bound obtained this way results from the fact that it considers the maximum possible number of aircraft service completions, as well as the minimum possible flight load requirements.

With the new flight load requirement definition, a particular aircraft combination can result in many different CFA_h levels, depending on the exact value of the total flight time in each time period of the planning horizon. Therefore, in order to search for the optimal solution, the user needs to consider alternative aircraft combinations, verify their feasibility, and check the optimal CFA_h they result in. This can be accomplished by enforcing the complete aircraft combination using the model formulation of Section 6.2.

6.5.2 Multi-squadron units

Another interesting problem extension arises when the considered unit is comprised of several sub-units. The incorporation of multi-squadron units adds a strong combinatorial flavor to our model, since distinct cases depending on which squadron an aircraft that enters or exits the maintenance station belongs to need to be distinguished. In turn, this has a significant impact on the computational requirements of the proposed algorithm, as well as on those of CPLEX/LINGO.

Note, however, that instead of considering each of these possible cases separately, we can simply check whether a particular CFA_h level can be attained by a feasible combination by utilizing the complete constraint set of problem (6.2.1)-(6.2.30) and letting the solver deduce whether an associated feasible flight and maintenance plan exists. The computational requirements that result when the CFA_h level is fixed to a particular value are moderate, enabling the solution of realistic problems in satisfactory computational times.

6.5.3 Alternative problem objectives

In the current chapter, we consider the cumulative residual flight time availability as the model's objective. One of the reasons for doing so is because among all the alternative objectives this is the one that causes the largest computational difficulties to the solution of the optimization model of Section 6.2. To justify this claim, in Table 6.13 we report the computational requirements of LINGO and CPLEX on the same problem instances as those of Table 6.6, except that we use the cumulative aircraft availability (CFA_a) instead of the cumulative residual flight time availability as the model's objective. Mathematically, this objective is expressed as follows:

$$\text{Max} \sum_{t=2}^{T+1} \sum_{n \in N} a_{n,t}$$

The corresponding single objective multi- period formulation (S-FMPa) is:

$$\text{Max } CFA_a \qquad \text{(S-FMP}_\text{a}\text{)}$$

$$\text{s.t. } x \in X$$

The results of Table 6.13 confirm that the use of the alternative objective reduces considerably the computational requirements of the two software packages, and suggest that the objective choice is a very crucial decision that has a strong impact on the computational effort needed to find the optimal solution of the problem. A similar behavior is observed when the minimum residual flight time and aircraft availability are used as model objectives, which are expressed as follows:

$$\text{Max } Z$$

$$\text{s.t. } Z \leq \sum_{n \in N} y_{n,t},\ t = 2,\ldots,T+1, \text{ and}$$

$$\text{Max } Z$$

$$\text{s.t. } Z \leq \sum_{n \in N} a_{n,t},\ t = 2,\ldots,T+1, \text{ respectively.}$$

The proposed algorithm requires a few modifications in this case, since maximizing the number of aircraft service completions is not necessarily an optimal maintenance strategy. In view of the above important insights, however, we did not pursue these modifications, because the performance of the existing solvers on realistic problems is quite satisfactory, and the potential savings from the implementation of a modification of the above algorithm are uncertain.

Table 6.13: Computational times (seconds) of LINGO/CPLEX for the alternative objective CFA_a

	LINGO		CPLEX single-thread	
$\|N\|$	**Avg**	**Max**	**Avg**	**Max**
10	0.12	0.17	0.02	0.03
15	0.19	0.72	0.02	0.03
20	0.32	2.39	0.03	0.05
25	0.22	1.26	0.03	0.05
30	0.25	0.69	0.04	0.06
50	0.44	1.05	0.04	0.05
100	1.45	15.36	0.08	0.08
200	2.20	2.78	0.14	0.16

Out of all these alternative model objectives, there does not seem to exist a single one that can capture completely all the aspects of the problem under consideration. For example, the CFA_h is not able to differentiate between solutions with different distribution of the total fleet availability among the time periods of the planning horizon, which is also important since this distribution should be as balanced as possible, too. We believe that the user would have to develop a multi-objective model in order to capture this problem aspect, but we did not pursue this because it would require additional modifications, and would extend the length of this chapter beyond the typical standards.

Our particular choice for the model objective was motivated by the observation that, in contrary to this objective, other ones can be handled successfully by commercial optimization software packages such as CPLEX/LINGO. Hence, given also the fact that this objective is quite realistic, we decided to develop a specialized algorithm that handles it in order to fill in this gap. Independently of that, for each typical alternative model objective we believe that it is possible to develop a suitable modification of the proposed algorithm that will be able to solve the problem under consideration. The main design of this algorithm will remain similar to that of the present one (find a valid upper bound on the optimal objective, test the feasibility of aircraft combinations that attain this bound, eliminate infeasible such combinations, update accordingly the upper bound, etc., until the optimal solution is obtained), although the particular details for the implementation of each single step will clearly have to be modified accordingly in each case. We do not claim that the performance of such an algorithm will always be superior to that of commercial optimization software (this cannot be known in advance), but that the development of such an algorithm appears attainable.

6.6 Summary

In this chapter, we addressed a single-objective multi-period version of the FMP problem. The problem aims to issue a joint flight and maintenance plan for a group of aircraft that comprise a unit in order to maximize the unit's fleet availability over a multi-period planning horizon. Using the cumulative residual flight time availability as the objective, we developed an exact solution algorithm that initially computes a valid upper bound on its optimal value, and then gradually reduces this bound in a stepwise fashion, until a solution that attains it is identified. The performance of the algorithm on realistic problems appears superior to that of two commercial optimization solvers that can be used alternatively for the solution of the problem, whereas the opposite behavior is observed for a class of problems with significantly different characteristics.

Chapter 7 Incorporating the Minimization of the Fleet Availability Variability

7.1 Introduction

A drawback of the FMP models studied in the previous chapters is that they provide long-term fleet availabilities with significant variability. For example, the application of the FMP model developed by Kozanidis et al. (2010) for solving a real problem instance in 4 consecutive planning horizons with an equal length of 6 monthly periods each, results in the fleet availability pattern shown in Figure 7.1. The characteristic bell-shaped curve of the fleet availability depicted in this figure is an indication of its high variability. Intuitively, the availability levels tend to be higher towards the middle of the associated planning horizon, and lower towards its two endpoints. This is a consequence of the fact that the model focuses on maximizing the fleet availability within each individual planning horizon in isolation, without taking into consideration the fact that the transition into the next planning horizon must also be as smooth as possible; this clearly results in plans which, although being optimal within each individual planning horizon, do not exhibit certain desirable long-term characteristics, such as low variability. An inevitable side effect of this behavior is the fact that the exact length of each planning horizon strongly affects the pattern of the fleet availability; the longer this length, the lower the associated variability is expected to be. Still, however, since the application of the model in subsequent rolling horizons is inevitable, the bell-shaped pattern depicted in Figure 7.1 is always expected to be present.

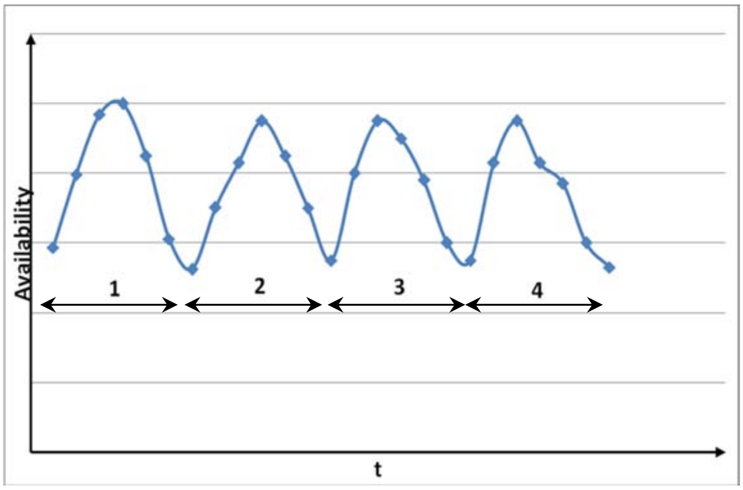

Figure 7.1: Visual depiction of the fleet availability's variability

With these in mind, we formulate a quadratic FMP model in this chapter, which, besides the typical objective maximizing the fleet availability, also includes an additional objective minimizing its variability. For this model, we develop two specialized solution algorithms, which successfully obtain the entire frontier of non-dominated solutions. Both algorithms utilize suitable relaxations of the original formulation, exploiting the discrete nature of the domain comprising possible fleet availability values. This key property is a consequence of the fact that the flight time availability of the unit is uniquely determined by the combination of aircraft entering and exiting the maintenance facility, and the fact that the number of distinct such combinations is finite.

The first methodology disaggregates the original FMP model into smaller subproblems whose solution is attained much more efficiently. Initially, it establishes a valid upper bound on the ideal value of the fleet availability through the solution of a suitable relaxation of the original formulation; next, this value is gradually reduced in a stepwise manner, with the aircraft combination minimizing the variability of the associated fleet availability identified in each of these steps. Adding special valid inequalities for excluding the solutions which cannot be optimal, the procedure succeeds in obtaining the entire frontier of non-dominated solutions upon termination.

The second methodology is a variant of the ε-constraint method, applied to a suitable relaxation instead of the original FMP model. It works by devising the payoff table calculation through lexicographic optimization, and by disaggregating the FMP solution into suitable relaxations which are utilized in subsequent steps. As the experimental results that we present demonstrate, the computational performance of the two proposed algorithms is considerably superior to that of applying the typical ε-constraint method directly on the original biobjective model, enabling the solution of large realistic problem instances in reasonable computational times.

The remainder of this chapter is structured as follows. In Section 7.2 we present the proposed FMP formulation. In Section 7.3, we develop the proposed solution methodologies, while in Section 7.4 we present computational results comparing their relative performance. Section 7.5 discusses some interesting model enhancements and possible extensions, and finally, Section 7.6 summarizes our conclusions.

7.2 Biobjective FMP model (Bi-FMPh)

In this section, we develop the mixed integer biobjective quadratic formulation for the FMP problem, which extends the single objective FMP formulation presented in Section 6.2. As already explained at the end of Section 6.2, that model can be expressed as follows:

$$\text{Max } CFA_h(x) \qquad \text{(S-FMP}_\text{h}\text{)}$$
$$\text{s.t. } x \in X \, ,$$

where x is a solution, $CFA_h(x)$ is the cumulative residual flight time availability realized by this solution, and X is the set of all feasible solutions.

Let $SY_t = \sum_{n=1}^{|N|} y_{n,t}$ be the individual residual flight time availability of the unit in time period t, and $\overline{SY}(x) = \frac{CFA_h(x)}{T} = \frac{\sum_{t=2}^{T+1} SY_t(x)}{T}$ be the average residual flight time availability of the unit over the planning horizon realized by solution x. We define the corresponding variability, $V_h(x)$, of this solution as:

$$V_h(x) = \left(\frac{1}{T}\right)\sum_{t=2}^{T+1}(SY_t(x) - \overline{SY}(x))^2 .$$

This expression is similar but not identical to the biased sample variance definition, since the sum of the squared deviations from the average is divided by the number of observations instead of the number of observations minus one. In our case, this difference is insignificant, since our main focus is on minimizing the summation of the squared deviations from the average. The incorporation of the second objective minimizing the variability of the residual flight time availability leads to the following biobjective formulation for the FMP problem (Bi-FMP$_h$):

$$\begin{aligned} &\text{Max } CFA_h(x) \\ &\text{Min } V_h(x) \\ &\text{s.t. } x \in X \end{aligned} \qquad \text{(Bi-FMP}_h\text{)}$$

A well-known notion in multi-objective optimization is that of *efficiency* or *Pareto optimality*. In the context of the specific formulation introduced above, a feasible solution x^* is called *efficient* or *Pareto optimal* if there does not exist any other feasible solution x' such that $CFA_h(x^*) \leq CFA_h(x')$ and $V_h(x^*) \geq V_h(x')$, with at least one of the two inequalities holding as strict. In general, efficient are the solutions for which one cannot improve one of the two objectives without worsening the other. In multi-objective optimization, the different objectives involved are typically in conflict, and very rarely a single solution that simultaneously optimizes all of them exists. We use the notation $z(x)$ to denote the image of a feasible solution in objective space: $z(x) = \{(CFA_h(x), V_h(x)) : x \in X\}$.

If x^* is efficient, then the corresponding point $z(x^*) = (CFA_h(x^*), V_h(x^*))$ in objective space is called *non-dominated point*. The set of all efficient solutions constitutes the *efficient set*, E, while the set of all non-dominated points constitutes the *non-dominated set*, N_d:

$$N_d = \{(CFA_h(x^*), V_h(x^*)) : x^* \in E\}.$$

Let $CFA_h^I = \max_{x \in X} CFA_h(x)$ and $V_h^I = \min_{x \in X} V_h(x)$. Then, the point $(CFA_h^I,\ V_h^I)$ in objective space is called the *ideal point*. In general, the ideal point is not a member of N_d, since a feasible solution in decision space whose corresponding image in criterion space is the ideal point rarely exists. Let also $V_h^N = \min_{x \in X} V_h(x) : CFA_h(x) = CFA_h^I$ and $CFA_h^N = \max_{x \in X} CFA_h(x) : V_h(x) = V_h^I$. The points $\left(CFA_h^N,\ V_h^I\right)$ and $\left(CFA_h^I,\ V_h^N\right)$ in objective space belong to N_d, while their inverse images in decision space belong to E. Finally, the point $(CFA_h^N,\ V_h^N)$ is called the *nadir point* and is rarely a member of N_d, because it is usually dominated by the two previous points.

7.3 Solution methodology

7.3.1 Theoretical groundwork

Let en_t and ex_t denote the number of aircraft entering and exiting, respectively, the maintenance facility at the beginning of time period t (t = 2,…,T+1), and $c = (en_2,\dots,en_{T+1},\ ex_2,\dots,ex_{T+1})$ be a specific such aircraft combination over the entire planning horizon. The two solution methodologies that we develop for the biobjective quadratic FMP problem utilize the key property that the domain comprising possible CFA_h values is a discrete set. This enables the identification of the entire non-dominated set through the application of an iterative solution procedure that minimizes the associated variability for each of these discrete CFA_h values. In that sense, our proposed approach exhibits some similarities with the methodology proposed by Bérubé et al. (2009), who develop an exact ε-constraint method for biobjective combinatorial optimization problems with integer objective values. Before introducing the specifics of this approach, we prove some important theoretical results first, which, among others, lead to the establishment of this key property, laying the foundation for the development of the proposed algorithmic methodologies.

Lemma 7.1: The maximization of CFA_h depends solely on the combination of aircraft entering and exiting the maintenance facility in time periods 2,…,T+1.

Proof. For t = 2,…,T+1, the residual flight time availability of the unit at the beginning of time period t, SY_t, is equal to $\sum_{n=1}^{|N|} Y1_n - \sum_{k=1}^{t-1} S_k + \sum_{k=2}^{t} (Y \cdot ex_k)$. The first two summations in this expression are constant; therefore, the only term subject to optimization is the last summation, which is equal to $(Y \cdot ex_2 + \dots + Y \cdot ex_t)$. As a result, maximizing CFA_h is equivalent to maximizing $\sum_{t=2}^{T+1} \sum_{k=2}^{t} (Y \cdot ex_k)$, which can be expressed as $(Y \cdot ex_2) + (Y \cdot ex_2 + Y \cdot ex_3) + \dots + (Y \cdot ex_2 + \dots + Y \cdot ex_{T+1}) = (T \cdot ex_2 + (T-1) \cdot ex_3 + \dots + ex_{T+1}) \cdot Y$.

Besides decision variables en_k and ex_k, which determine the combination of aircraft entering and exiting the maintenance facility in time periods 2,…,T+1, the above expression only includes Y and T, which are fixed parameters known in advance. This establishes the validity of the lemma.

Lemma 7.2: The CFA_h values of any two distinct aircraft combinations differ by an exact multiple of Y.

Proof. In Lemma 7.1, it was shown that the CFA_h value of a particular aircraft combination is equal to $T\sum_{n=1}^{|N|} Y1_n - \sum_{t=1}^{T}(T-t+1)S_t + (T\cdot ex_2 + (T\text{-}1)\cdot ex_3 + \ldots + ex_{T+1})\cdot Y$. In this expression, the term $T\sum_{n=1}^{|N|} Y1_n - \sum_{t=1}^{T}(T-t+1)S_t$ is constant and independent of the aircraft combination. Besides, the multiplier of Y, appearing in the last parenthesis, is clearly an integer; the fact that the difference of any two integers is also an integer establishes the validity of the lemma.

Corollary 7.1: The number of feasible aircraft combinations $c = (en_2,\ldots,en_{T+1},ex_2,\ldots,ex_{T+1})$ is finite.

Lemma 7.3: Minimizing V_h for a fixed CFA_h level is equivalent to minimizing $SY_2^2 + SY_3^2 + \ldots + SY_{T+1}^2$.

Proof. For a fixed CFA_h level, we have $V_h = \left(\frac{1}{T}\right)\sum_{t=2}^{T+1}(SY_t - \overline{SY})^2 =$

$$\frac{1}{T}\left(\left(SY_2^2 + \overline{SY}^2 - 2SY_2\cdot\overline{SY}\right) + \left(SY_3^2 + \overline{SY}^2 - 2SY_3\cdot\overline{SY}\right) + \ldots + \left(SY_{T+1}^2 + \overline{SY}^2 - 2SY_{T+1}\cdot\overline{SY}\right)\right) =$$

$$\frac{1}{T}\left(SY_2^2 + SY_3^2 + \ldots + SY_{T+1}^2 + T\cdot\overline{SY}^2 - 2\overline{SY}\left(SY_2 + SY_3 + \ldots + SY_{T+1}\right)\right) =$$

$$\frac{1}{T}\left(SY_2^2 + SY_3^2 + \ldots + SY_{T+1}^2 + T\left(\frac{CFA_h}{T}\right)^2 - 2\frac{CFA_h}{T}\left(CFA_h\right)\right) =$$

$$\frac{1}{T}\left(SY_2^2 + SY_3^2 + \ldots + SY_{T+1}^2 - \frac{CFA_h^{\,2}}{T}\right) = \left(\frac{SY_2^2 + SY_3^2 + \ldots + SY_{T+1}^2}{T}\right) - \left(\frac{CFA_h}{T}\right)^2.$$

Since CFA_h is fixed in this expression, the only non-constant quantity subject to optimization is the enumerator of the first term. This establishes the validity of the lemma.

In the next sections, we utilize the above key theoretical results for the development of the proposed solution methodologies.

7.3.2 ε-Constraint methodological framework

An appropriate methodology for the solution of a biobjective problem should ideally be able to provide the entire set of non-dominated solutions to the decision maker, allowing him/her to make the final decision on the desired compromise between the two objectives. Since E is defined in the decision space while N_d is defined in the objective space, in the general case there may be multiple efficient solutions that map to the same non-dominated point. For those cases, we make the reasonable assumption that identifying one of these solutions is sufficient. This implies that no other characteristic other than the two associated objective values can be used for ranking any two alternative solutions, an assumption which appears reasonable and valid.

One of the most commonly used solution approaches in multi-objective programming is the weighted-sums method (Ehrgott, 2005), which optimizes a suitable convex combination of the model objectives. A natural difficulty that arises in our case is that the specification of the weights, which strongly influences the results, is not straightforward. Moreover, it is well known that the method is unable to find a certain class of efficient solutions (namely, unsupported ones) in discrete optimization problems (Ehrgott and Gandibleux, 2000), and that it does not provide any guarantee that it will not overlook efficient solutions in the general case.

An alternative method which, under suitable treatment, can generate the non-dominated set of non-convex optimization problems is the ε-constraint reduced feasible region method (Cohon, 1978; Chankong and Haimes, 1983). This method optimizes a single one of the objectives, while transforming all the other ones into constraints through the imposition of suitable bounds on their values. These bounds are lower for maximization objectives and upper for minimization ones. Taking advantage of the fact that the domain comprising feasible CFA_h values is a discrete set, we model the corresponding ε-constraint formulation for the Bi-FMP_h problem as follows:

$$\begin{gathered}\text{Min } V_h(x)\\ \text{s.t. } CFA_h(x) \geq \varepsilon\\ x \in X\end{gathered}$$

The traditional ε-constraint method may generate inefficient solutions (e.g., see Xidonas et al., 2010). In order to remedy this well-known pitfall, Mavrotas (2009) has proposed a method called AUGMECON, which employs an acceleration algorithm of early exit in order to avoid the generation of dominated solutions. Mavrotas and Florios (2013) developed an improved extension of this method (AUGMECON2), which introduces a bypass coefficient of the innermost loop. Another variant of this approach (SAUGMECON) that extends the early exit acceleration algorithm with bouncing steps was more recently developed by Zhang and Reimann (2014). In the present chapter, we employ AUGMECON2 for the solution of the problem under investigation. In summary, this method consists of the following steps:

Step 1 :

Select one as the main objective subject to optimization, and convert the other (secondary) objective(s) into constraints. Then, create the payoff table by lexicographic optimization of the objective functions. The range of each objective is determined by the corresponding interval between its ideal and its nadir value.

Step 2 :

Divide the range of each secondary objective function to m equal intervals using m-1 intermediate equidistant grid points (ε_k) which are used to vary parametrically the associated right-hand side.

Step 3 :

Solve the augmented ε-constraint model for each ε_k value obtained in Step 2.

The procedure outlined above can be adjusted accordingly to generate the entire frontier of non-dominated solutions in our case, taking advantage of the discrete nature of the CFA_h objective. As Mavrotas and Florios (2013) have shown, the solution obtained upon the completion of any iteration is guaranteed to be non-dominated. This key result motivated our decision to utilize the AUGMECON2 method for the solution of the problem, since it ensures that no redundant iterations will be performed. Following the guidelines proposed by Mavrotas and Florios (2013), the ε-constraint optimization model is formulated as follows in our case:

$$\begin{aligned} &\text{Min } V_h(x) - eps \cdot \frac{s}{r} \\ &\text{s.t. } CFA_h(x) - s = \varepsilon_k \\ &\quad x \in X, \end{aligned} \qquad \text{AUGMECON Bi-FMP}_\text{h}$$

where eps is a sufficiently small scalar ensuring the lexicographic ordering of the two objectives, s is a slack variable utilized to convert the constraint imposing the lower bound on the CFA_h objective into an equality, and r is the range of possible CFA_h values. Based on the theory developed in the previous section, and in particular Lemmas 7.1, 7.2 and Corollary 7.1, it is obvious that starting from the nadir CFA_h value and increasing it by a step size of Y at each iteration, we can consider all the ε_k values which differ from it by an exact multiple of Y, eventually stopping when the ideal CFA_h value is reached. This way, all possible CFA_h values will be examined and the entire non-dominated set will be obtained upon termination.

7.3.3 Double-step solution algorithm

Step 1 of the AUGMECON2 method, and in particular the creation of the payoff table by lexicographic optimization, necessitates the separate solution of one single-objective optimization problem for each of the model's objective functions. In the previous chapter, it was shown that performing this task with commercial optimization solvers in reasonable computational times is only possible for problem instances with up to 25-30 aircraft. A typical aircraft wing of the HAF, on the other hand, may consist of up to 60-100 aircraft. The exact solution algorithm developed in the previous chapter is far more efficient, enabling the successful treatment of such size problems in reasonable computational times. Motivated by this, in what follows we slightly modify the design of this algorithm in order to enable its applicability for the efficient treatment of the biobjective version of the problem, too.

The proposed solution algorithm identifies initially a valid upper bound on the ideal CFA_h value by solving a simplified relaxation of the original model formulation; in successful iterations, this value is gradually decreased by a constant value of Y. For each distinct CFA_h level encountered, the algorithm tries to identify the minimum variability associated solution. Due to the fact that the computational requirements of utilizing the original formulation for performing this task are excessive, a suitable relaxation of the original model is used instead to this end. Each identified solution is checked for full feasibility, by adjusting the original formulation accordingly to impose its realization. Valid cuts excluding infeasible solutions from further consideration are suitably added, enabling the continuation of the search for the optimal solution to a given CFA_h level.

For each individual CFA_h level, the procedure terminates either when the optimal solution realizing this CFA_h level is identified, or when it is proven that no such feasible solution exists. In the former case, the associated point (CFA_h, V_h) in objective space is added to the set of points which are candidate for being non-dominated. All the cuts that pertain to a specific CFA_h level remain active for as long as this level remains fixed. Once the search at the current CFA_h level terminates, these cuts are rendered redundant and are subsequently suppressed, while the algorithm proceeds to the next lower CFA_h level.

The algorithmic procedure continues in a similar fashion until all possible CFA_h levels have been considered; at that point, a straightforward approach identifies among all the points which have been recorded those which are non-dominated, together with their efficient solution counterparts. The following subsections portray in thorough detail the steps of the double-step solution algorithm.

7.3.3.1 Bounding the ideal CFA_h

Due to constraint set (6.2.8), the unit residual flight time availability of each time period is reduced by the associated flight load requirements independently of the exact flight time of each individual aircraft. As a result, the maximum possible CFA_h value is attained when each grounded aircraft finishes its service as early as possible. This is true because interrupting the maintenance service of a grounded aircraft once it has begun could potentially delay the addition of this aircraft's phase interval to the unit fleet availability. This would clearly reduce the CFA_h value leading to a sub-optimal solution, since the objective function weighs more heavily the number of aircraft exiting the facility at any particular time period than that of any succeeding one. Of course, if the maintenance facility does not have sufficient time capacity, the maintenance service of a grounded aircraft may span several time periods.

In order to compute a valid upper bound on the ideal CFA_h value, we impose the maximum possible aircraft flow into the maintenance facility, by assigning the flight load of each time period to the available aircraft in non-decreasing order of their residual flight times. As a consequence, each available aircraft is grounded for service at the earliest possible time. To speed up the performance of the algorithm, not all the original model constraints are considered for calculating this upper bound, which indicates that the corresponding solution that will be identified may be infeasible. This necessitates a consequent check that confirms or disproves the feasibility of this solution.

Procedure CFA_h-UB, presented in Section 6.3.1, can be utilized for computing the valid upper bound on the ideal CFA_h value. Additionally, this procedure also identifies a specific combination $c^{\text{nom}} = (en_2^{\text{nom}},\ldots,en_{T+1}^{\text{nom}}, ex_2^{\text{nom}},\ldots,ex_{T+1}^{\text{nom}})$ of aircraft entering and exiting the maintenance facility in each time period. In what follows, we call this combination *nominal*. Note that in Section 6.3.1 we showed that, for $t = 2,\ldots,T+1$, the total number of aircraft exiting the maintenance facility from time period 2 up to time period t for $t = 2,\ldots,T+1$ in any feasible solution cannot be larger than $\sum_{k=2}^{t} ex_k^{\text{nom}}$, while the total number of aircraft entering the maintenance facility from time period 2 up to time period t for $t = 2,\ldots,T+1$ in any feasible solution cannot be larger than $\sum_{k=2}^{t} en_k^{\text{nom}}$.

7.3.3.2 Checking a particular CFA_h level for feasibility

Each specific CFA_h level considered can be optionally checked for feasibility first. This is carried out by checking whether a feasible solution to the original formulation (6.2.2)-(6.2.30) after fixing this CFA_h value exists. As it turns out, the computational requirements for performing this are limited, due to the fact that knowing in advance the CFA_h level eliminates considerably the combinatorial nature of the problem. The execution of this step is optional, however, since the feasibility of the current CFA_h level is always confirmed or disproved at the next step, in which the minimum variability aircraft combination is sought. Nevertheless, our computational experience suggests that this step should preferably be executed, because it guards against the risk of visiting an exponential number of infeasible aircraft combinations before realizing that a particular CFA_h level is infeasible.

7.3.3.3 Obtaining the minimum variability aircraft combination

For each CFA_h level considered, the algorithm tries to find the minimum variability aircraft combination realizing this CFA_h level. Due to the substantial computational effort required for performing this, a suitable relaxation of the original formulation is utilized to this end. The main characteristic of this relaxation is that it does not include distinct decision variables monitoring the residual flight and maintenance time of each individual aircraft; instead, it utilizes decision variables monitoring the cumulative flight and maintenance times of all aircraft for each time period of the planning horizon. This clever technique results in a considerable reduction of the number of decision variables utilized, making it possible to bypass the computational difficulties involved.

To give a specific example, exact knowledge of the individual aircraft residual flight and maintenance times determines the exact number of available and grounded aircraft in each time period. In the relaxed formulation, however, this knowledge is absent. Therefore, the model estimates the number of available and grounded aircraft in each time period using suitable bounds on the cumulative residual flight and maintenance times of all aircraft.

Due to the adoption of several simplifications such as the above, a particular aircraft combination identified may turn out to be infeasible. Therefore, a proper feasibility check is performed on each such combination, using the full original FMP formulation. Despite the extra effort needed to perform this task, the proposed relaxation facilitates considerably the identification of the minimum variability solution that realizes a particular CFA_h level. The following additional mathematical notation is utilized in this formulation:

Parameters:

ε : a sufficiently small number,

Decision variables:

w_t : binary decision variable equal to 1 if the time capacity of the maintenance facility in time period t ($t = 1,\dots,T$) is greater or equal to the total residual maintenance time of all grounded aircraft at the beginning of the same time period, and 0 otherwise,

$n\alpha_t$: number of grounded aircraft at the beginning of time period t ($t = 2,\dots,T+1$),

$LBrem_t$: lower bound on the number of aircraft that remain grounded at the end of time period t ($t = 1,\dots,T$),

SG_t : total residual maintenance time of all grounded aircraft at the beginning of time period t, ($t = 2,\dots,T+1$),

$SGres_t$: total residual maintenance time of all grounded aircraft at the end of time period t ($t = 1,\dots,T$).

Utilizing the above notation and exploiting Lemma 7.3, we employ the following mixed integer quadratic formulation in order to identify the minimum variability aircraft combination, $c = (en_2,\dots,en_{T+1}, ex_2,\dots,ex_{T+1})$, attaining a particular CFA_h level:

$$\text{Min } SY_2^2 + SY_3^2 + \dots + SY_{T+1}^2 \quad (7.3.1)$$

$$\text{s.t } T\sum_{n=1}^{|N|} Y1_n - \sum_{t=1}^{T}((T-t+1)S_t) + Y \cdot \sum_{t=2}^{T+1}((T-t+2)ex_t) = CFA_h \quad (7.3.2)$$

$$\sum_{k=2}^{t} ex_k \le \sum_{k=2}^{t} ex_k^{\text{nom}},\ t = 2,..,T+1 \quad (7.3.3)$$

$$\sum_{k=2}^{t} en_k \leq \sum_{k=2}^{t} en_k^{\text{nom}},\ t = 2,..,T+1 \tag{7.3.4}$$

$$\sum_{n \in N: Y1_n > 0} \min(Y1_n, t \cdot X_{\max}) + \sum_{k=2}^{t} (ex_k \cdot \min(Y, (t-k+1) \cdot X_{\max})) \geq \sum_{k=1}^{t} S_k \ \ ,\ t=1,..,T \tag{7.3.5}$$

$$SG_t \leq B_t + K(1-w_t),\ t=1,..,T \tag{7.3.6}$$

$$B_t \leq SG_t - \varepsilon + Kw_t,\ t=1,..,T \tag{7.3.7}$$

$$LBrem_t \leq C \cdot (1-w_t),\ t=1,..,T \tag{7.3.8}$$

$$\frac{(SG_t - B_t)}{G} \leq LBrem_t \leq \frac{(SG_t - B_t)}{G} + 1 - \varepsilon + K \cdot w_t,\ t=1,..,T \tag{7.3.9}$$

$$na_t \leq ex_{t+1} + K(1-w_t),\ t=1,..,T \tag{7.3.10}$$

$$ex_{t+1} \leq na_t - LBrem_t,\ t=1,..,T \tag{7.3.11}$$

$$SG_t - B_t - Kw_t \leq SGres_t \leq SG_t - B_t + Kw_t,\ t=1,..,T \tag{7.3.12}$$

$$SGres_t \leq K(1-w_t),\ t=1,..,T \tag{7.3.13}$$

$$SG_{t+1} = SGres_t + G \cdot en_{t+1},\ t=1,..,T \tag{7.3.14}$$

$$SY_t = SY_{t-1} - S_{t-1} + Y \cdot ex_t,\ t=2,..,T+1 \tag{7.3.15}$$

$$na_t = na_{t-1} + en_t - ex_t,\ t=2,..,T+1 \tag{7.3.16}$$

$$na_t \leq C,\ t=2,..,T+1 \tag{7.3.17}$$

$$SY_1 = \sum_{n=1}^{|N|} Y1_n \tag{7.3.18}$$

$$SG_1 = \sum_{n=1}^{|N|} G1_n \tag{7.3.19}$$

$$na_1 = \sum_{n=1}^{|N|} (1 - A1_n) \tag{7.3.20}$$

$$SG_t \geq 0,\ t = 2,..,T+1 \tag{7.3.21}$$

$$SGres_t \geq 0,\ t=1,..,T \tag{7.3.22}$$

$$en_t, ex_t, na_t \in Z^+,\ t = 2,..,T+1 \tag{7.3.23}$$

$$LBrem_t \in Z^+,\ w_t \ binary,\ t=1,..,T \tag{7.3.24}$$

The objective (7.3.1) minimizes the variability of the aircraft combination that will be identified, as a consequence of Lemma 7.3 and the fact that the CFA_h level is fixed at a specific value by constraint (7.3.2). For $t = 2,\ldots,T+1$, constraint sets (7.3.3) and (7.3.4) impose upper bounds on the cumulative number of aircraft exiting and entering the maintenance facility, respectively, from the beginning of the planning horizon up until time period t, as determined by the nominal combination. These two constraint sets are identical to constraint sets (6.3.2) and (6.3.3), respectively, of the model presented in Section 6.3.3.

For $t = 1,\ldots,T$, constraint set (7.3.5) ensures that the cumulative flight load of time periods 1 to t is satisfied. The required flight time to this end is provided by the aircraft which are available in time period 1 and by those that exit the maintenance facility in time periods $2,\ldots,t$. The flight time that each aircraft can provide in a single time period cannot be larger than the minimum between its residual flight time at the beginning of the same time period and X_{max}.

Constraint sets (7.3.6) and (7.3.7) ensure that binary variable w_t becomes equal to 1 when $SG_t \leq B_t$, and 0 otherwise. This variable is used to impose the restriction that no part of the maintenance time capacity should go wasted unless all grounded aircraft finish their service. For $t = 1,\ldots,T$, constraint sets (7.3.8) and (7.3.9) give proper value to $LBrem_t$ according to the corresponding value of w_t. If $w_t = 1$, then constraint set (7.3.8) ensures that $LBrem_t$ is equal to 0, while constraint set (7.3.9) becomes redundant, since $\frac{SG_t - B_t}{G}$ is non-positive. If $w_t = 0$, then constraint set (7.3.8) becomes redundant, while constraint set (7.3.9) ensures that $LBrem_t$ is equal to the ceiling of $\frac{SG_t - B_t}{G}$, which is a valid lower bound on the number of aircraft that will remain grounded at the end of time period t. For example, consider the hypothetical case of four grounded aircraft with residual maintenance time 20 hours each, for which $G = 320$ and $B_t = 20$. Independently of the individual aircraft maintenance times, at least 3 aircraft will remain grounded at the end of this time period. The computed lower bound is equal to $\left\lceil \frac{80-20}{320} \right\rceil = 1$, which is clearly valid. Due to the fact that in the relaxed formulation there is no knowledge of the individual aircraft maintenance times, this bound will not always be tight, as happens in this case. If the actual residual maintenance time distribution among the 4 aircraft were 5, 5, 5 and 65 instead, it would be possible to finish the service of 3 aircraft in the current time period, in which case the bound would be tight.

Constraint sets (7.3.10) and (7.3.11) impose suitable bounds on the number of aircraft exiting the maintenance facility at the beginning of each time period. If $w_t = 1$, then these constraints are equivalent to $ex_{t+1} = na_t$, which is valid since this implies that all grounded aircraft will complete their maintenance service. If w_t = 0, then constraint (7.3.10) becomes redundant, while constraint set (7.3.11) ensures than no more than $(na_t - LBrem_t)$ aircraft will exit the maintenance facility at the beginning of time period t+1, which is also valid according to the definition of $LBrem_t$.

Constraint sets (7.3.12), (7.3.13) and (7.3.14) update SG_{t+1} based on the corresponding value of w_t. If w_t = 0, then SG_{t+1} is set equal to $SG_t - B_t + G \cdot en_{t+1}$ by constraints (7.3.12) and (7.3.14), while constraint (7.3.13) becomes redundant. If w_t = 1, then SG_{t+1} is set equal to $G \cdot en_{t+1}$ by constraints (7.3.13) and (7.3.14), while constraint (7.3.12) becomes redundant. Constraint sets (7.3.15) and (7.3.16) update the unit residual flight time availability and the number of grounded aircraft, respectively, in each time period, while constraint (7.3.17) ensures that the latter will never exceed the maintenance facility's space capacity, similarly to the constraint set (6.3.5) of the model presented in Section 6.3.3. Constraint sets (7.3.18)-(7.3.20) initialize the unit residual flight time availability, the total residual maintenance time of the grounded aircraft, and the number of grounded aircraft, respectively, at the beginning of the planning horizon. Finally, constraint sets (7.3.21)-(7.3.22) impose non-negativity constraints on the continuous decision variables, while constraint sets (7.3.23)-(7.3.24) impose integrality constraints on the discrete decision variables.

We conclude this subsection with the proof that, for a fixed CFA_h level, constraint set (7.3.2)-(7.3.24) comprises a relaxation of the original formulation, and thus for every feasible solution to the original problem there exists a corresponding feasible solution to this formulation. This confirms the validity of utilizing model (7.3.1)-(7.3.24) instead of the original one in order to identify the minimum variability aircraft combination for a particular CFA_h level. Our choice to adopt this strategy is motivated by the fact that the computational requirements of the relaxed formulation are significantly lower than those of the original one.

Proposition 7.1: Suppose that for a feasible solution to the original set of constraints (6.2.2)-(6.2.30) we calculate the number of aircraft entering (en_t) and exiting (ex_t) the maintenance facility, the number of grounded aircraft (na_t), and the total residual maintenance time of all grounded aircraft (SG_t) in each time period of the planning horizon. Then, this solution is feasible to constraint set (7.3.2)-(7.3.24), in the sense that there exist consistent values for all the decision variables of that formulation that satisfy it.

Proof. Constraint (7.3.2) fixes the CFA_h level at its corresponding value. Constraints (7.3.3) and (7.3.4) hold for any feasible solution to (6.2.2)-(6.2.30) as shown in the previous chapter. Adding side by side constraints (6.2.8) for k = 1,…,t, we obtain $\sum_{k=1}^{t}\sum_{n=1}^{|N|} x_{n,k} = \sum_{k=1}^{t} S_k$. The left hand side of this expression includes the cumulative flight time of all aircraft in time periods 1,…,t. For each initially available aircraft, n, the maximum such flight time before it is grounded is equal to $\min(Y1_n, t \cdot X_{max})$, while for any aircraft exiting the maintenance facility in time period k ($2 \leq k \leq t$), it is equal to min(Y, $(t\text{-}k+1)\cdot X_{max}$). As a result, constraints (7.3.5) are satisfied, too. For each time period t, constraints (7.3.6)-(7.3.14) update the decision variables pertaining to the maintenance facility, i.e., the total residual maintenance time of the grounded aircraft, the number of aircraft that remain grounded, and the number of aircraft that exit the maintenance facility. If $SG_t > B_t$, then these constraints set the number of aircraft that remain grounded at the end of time period t equal to at least $\left\lceil \frac{(SG_t - B_t)}{G} \right\rceil$. This is true due to constraints (6.2.11) and (6.2.12) of the original model, which ensure that no part of the maintenance time capacity is allowed to go wasted when $SG_t > B_t$. If $SG_t \leq B_t$ then w_t is set equal to 1, all grounded aircraft exit the maintenance facility at the beginning of time period t+1, and SG_{t+1} is set equal to $G\cdot en_{t+1}$. Otherwise, SG_{t+} is set equal to SG_t - B_t + $G\cdot en_{t+1}$. This is true due to constraints (6.2.5)-(6.2.7) of the original model, which update the residual maintenance time of each grounded aircraft. Constraints (7.3.15)-(7.3.17) update the unit's residual flight time availability and the number of grounded aircraft in each time period, and are clearly satisfied in the original problem, too. Finally, the initialization constraints (7.3.18)-(7.3.20) and the non-negativity/integrality constraints (7.3.21)-(7.3.24) clearly hold in the original formulation, too.

7.3.3.4 Checking a particular aircraft combination for feasibility

The check of whether a specific aircraft combination is feasible is straightforward. The important property that the optimal solution can always be identified even if a steady rotation of aircraft into and out of the maintenance station is preserved is valid for this problem, too. We utilize this important result, which was proven in Section 5.2 for the single period FMP model formulation, in order to simplify the procedure for checking the feasibility of a particular aircraft combination. Since the proof carries over practically unchanged to the present biobjective multi-period version of the problem, we do not repeat it here for brevity.

As a consequence of the above discussion, checking a specific aircraft combination for feasibility can be accomplished by checking whether a feasible solution that realizes this combination exists. To do this, we fix the decision variable values unambiguously determined by this combination, and we check the original model for feasibility. This simplification eliminates considerably the combinatorial nature of the optimization problem, enabling the completion of the feasibility check in negligible time.

7.3.3.5 Generating a cut for the exclusion of a particular aircraft combination

When a specific aircraft combination identified by model (7.3.1)-(7.3.24) is proven infeasible, continuing the search for a different aircraft combination that attains the current CFA_h level necessitates the addition of a valid cut excluding this combination from further consideration. Let the superscript *excl* refer to the specific combination that we wish to exclude. A suitable cut that achieves this is the following: $\sum_{t=2}^{T+1}|en_t^{excl} - en_t| + \sum_{t=2}^{T+1}|ex_t^{excl} - ex_t| \geq 1$. This constraint imposes the restriction that the number of aircraft exiting and entering the maintenance facility in any subsequent combination yet to be found by the algorithm should differ from the one being excluded in at least one time period. Of course, one needs to add one such cut for each infeasible aircraft combination. The nonlinearities present in this cut can be straightforwardly eliminated through the clever reformulation procedure presented in the previous chapter.

7.3.3.6 Stopping conditions and selection of non-dominated solutions

Since the payoff table is not calculated explicitly in the case of the Double-step solution algorithm, no nadir point is obtained. Therefore, the algorithm terminates when a suitable lower bound on CFA_h is reached. Since CFA_h can be expressed as: $\sum_{t=2}^{T+1}(\sum_{n=1}^{|N|}Y1_n - \sum_{k=1}^{t-1}S_k + \sum_{k=2}^{t}(Y \cdot ex_k))$, which is equal to $T\sum_{n=1}^{|N|}Y1_n - \sum_{t=1}^{T}((T-t+1)S_t) + \sum_{t=1}^{T}((T-t+1)Y \cdot ex_{t+1})$, an obvious lower bound on CFA_h is $T\sum_{n=1}^{|N|}Y1_n - \sum_{t=1}^{T}((T-t+1)S_t)$.

An additional stopping rule that can be employed is to terminate the algorithmic execution in case a solution with $V_h = 0$ is obtained, utilizing the trivial result that the variability is always non-negative. Since the algorithm begins with the highest possible CFA_h value and reduces it in a stepwise manner in succeeding iterations, the above two stopping rules ensure that no non-dominated solution will ever be overlooked. Upon termination, a simple search procedure distinguishes the non-dominated points among all the points which have been recorded.

7.3.4 Hybrid approach

In order to further enhance the performance of the proposed solution algorithm, we develop a hybrid double-step ε-constraint approach, which attempts to combine the computational advantages of AUGMECON2 with those resulting from the disaggregation of the original FMP model as shown in the previous section. Let Π be the set of all feasible aircraft combinations determined by constraint set (7.3.2)-(7.3.24). Note that, for any combination c, $V_h(c)$ is directly determined by (7.3.15) and (7.3.18). The idea is to apply the AUGMECON2 method not to the full original FMP model, but to this FMP relaxation instead. Mathematically, this can be expressed as follows:

$$\begin{aligned} &\text{Min } V_h(c) - eps \cdot \frac{s}{r} \\ &\text{s.t. } CFA_h(c) - s = \varepsilon_k \\ &\qquad c \in \Pi \end{aligned} \qquad \text{(HYBRID-FMP)}$$

Note that, in contrast to the Double-step solution method, constraint (7.3.2) merely provides the definition of CFA_h in this case, and does not fix the CFA_h level to a specific value. In summary, the proposed methodology includes the following steps:

Step 1 :
Select V_h as the main objective function to be optimized, and transform the secondary objective CFA_h into a constraint. Then, create the payoff table by lexicographic optimization of the objective functions, using the relaxed formulation. The range of the secondary objective function is determined by the corresponding interval between its ideal and its nadir value.

Step 2 :
Divide the range of the CFA_h objective to m equal intervals using m-1 intermediate equidistant grid points (ε_k) for varying parametrically the right-hand side of the corresponding constraint, which differ from each other by a constant value of Y.

Step 3 :
For each ε_k value obtained in Step 2, solve the augmented ε-constraint relaxed model and check the identified aircraft combination for feasibility by forcing the original formulation to realize it (same as 7.3.3.4). If this combination is infeasible, add a valid cut excluding it from further consideration (same as 7.3.3.5), and repeat until either a feasible combination is identified, or the current CFA_h level is proven infeasible.

The main difference in this case is that the creation of the payoff table by lexicographic optimization is attained efficiently even for realistic problem instances, due to the fact that the utilized formulation is a relaxation of the original FMP model. The computed objective ranges may initially be larger than those of the original problem since the relaxed model may also identify infeasible solutions, but these solutions are quickly rejected through the application of the feasibility check. Note that, since Step 3 is applied using a relaxation, the actions utilized by the Double-Step solution algorithm in order to identify feasible solutions to the original problem need to be employed in this case, too. Despite the extra effort needed to perform this, the performance of the algorithm appears considerably improved in comparison to the case that AUGMECON2 method is applied to the full original FMP model, as demonstrated by the computational results that we present in Section 7.4.

7.3.5 A Small Case Study

In this section, we illustrate the application of the proposed solution methodologies on a small case study. We consider a unit comprised of 10 aircraft. At the first period of the planning horizon, 9 aircraft are available and one is grounded. Table 7.1 presents the residual flight times of the available aircraft in plain-style, and the residual maintenance time of the grounded aircraft in bold-style. Table 7.2 presents the flight requirements and the time capacity of the maintenance facility in each time period. The values of the remaining problem parameters are Y = 300 hours, Y_{min} = 0.1 hours, G = 320 hours, G_{min} = 0.1 hours, C = 2, and X_{max} = 50. Collective results pertaining to the application of the three solution approaches are presented in Table 7.6.

Table 7.1: Residual flight times ($y_{n,1}$) / residual **maintenance time** ($\mathbf{g}_{n,1}$) (hours)

$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$\boldsymbol{n = 10}$
5	42	81	88	169	183	205	260	288	**253**

Table 7.2: Flight load requirements and time capacity of the maintenance facility

t	1	2	3	4	5	6
S_t	144	122	111	131	128	131
B_t	166	194	160	197	156	192

7.3.5.1 AUGMECON2 Methodology

Table 7.3 presents the payoff table for this problem. Since the range of the objective function CFA_h is [7328, 7928], we choose a step size of Y = 300 and we divide this range into two equal intervals; the three corresponding grid points for the ε-constraint method are ε_1 = 7328, ε_2 = 7628, and ε_3 = 7928. Next, the augmented ε-constraint model (AUGMECON Bi-FMP$_h$) is solved for each of these three values.

Table 7.3: Payoff table for the small case study

	V_h	CFA_h
Min V_h	6778.222	7328
Max CFA_h	9211.556	7928

7.3.5.2 Double-step solution algorithm

Given that each aircraft can fly at most X_{max} = 50 hours during each time period, the following table shows the maximum cumulative flight time (*MCFT*) that the aircraft which are available at the beginning of the planning horizon can provide from the beginning of the planning horizon up to time period t for $t = 1,\ldots,T$, before being grounded for service.

Table 7.4: Maximum cumulative flight times that can be provided by the initially available aircraft

t	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	*MCFT*
1	5	42	50	50	50	50	50	50	50	397
2	5	42	81	88	100	100	100	100	100	716
3	5	42	81	88	150	150	150	150	150	966
4	5	42	81	88	169	183	200	200	200	1168
5	5	42	81	88	169	183	205	250	250	1273
6	5	42	81	88	169	183	205	260	288	1321

The valid upper bound on the ideal CFA_h value provided by Procedure CFA_h-*UB* is equal to 7928. The corresponding (nominal) aircraft combination identified is the one shown in Table 7.5.

Table 7.5: Nominal aircraft combination identified by Procedure CFA_h-*UB*

t	2	3	4	5	6	7
en_t^{nom}	0	1	0	1	0	1
ex_t^{nom}	0	1	0	1	0	1

After setting CFA_h equal to 7928, model (7.3.1)-(7.3.24) also identifies the nominal combination, for which $V_h = 9211.556$. Next, we check whether this combination can be realized by a feasible solution. In order to do this, we retain the order of aircraft visiting and exiting the maintenance facility constant, and we fix the decision variables values determined by this combination. These variables are $a_{n,t}$, $d_{n,t}$, $f_{n,t}$, $p_{n,t}$, $r_{n,t}$, and q_t for $n = 1,\ldots,6$, $t = 2,\ldots,7$, (for example, $\alpha_{1,2} = 1$, $\alpha_{1,3} = 0$, $\alpha_{10,2} = 0$, $\alpha_{10,3} = 1$, etc.), together with several of the variables $x_{n,t}$, $y_{n,t}$, $g_{n,t}$, $h_{n,t}$. The user may choose to fix only a proper subset of these values and let the remaining ones be deduced by the solver, or opt for a more specific model by explicitly fixing all the unambiguously determined decision variable values. For our small case study, the resulting formulation is feasible. Therefore, no valid-cut excluding this combination needs to be added, and the associated point (7928, 9211.56) in objective space is added to the set of points which are candidate for being non-dominated.

The algorithm continues similarly with the next lower CFA_h levels, until the lower bound on CFA_h is reached. This bound is equal to $T\sum_{n=1}^{|N|} Y1_n - \sum_{t=1}^{T}((T-t+1)S_t) = 6728$. Upon termination, the full non-dominated set is obtained by eliminating the dominated solutions out of those that have been recorded.

7.3.5.3 Hybrid algorithm

The Hybrid algorithm creates initially the payoff table by lexicographic optimization of the objective functions. Although the relaxed formulation is utilized to this end, the ranges of the objectives coincide with those identified in the application of the AUGMECON2 method in Subsection 7.3.5.1. Next, the HYBRID-FMP model is employed using $\varepsilon_1 = 7328$, $\varepsilon_2 = 7628$ and $\varepsilon_3 = 7928$. The variability of the solution identified for $\varepsilon_1 = 7328$ is equal to 6778.222. The feasibility check applied next confirms that the associated aircraft combination is feasible. Thus, the corresponding point (7328, 6778.222) in objective space is added to the non-dominated set. The algorithm continues in a similar fashion with the next two higher CFA_h levels.

7.3.5.4 Collective results

The non-dominated set obtained from the application of the three solution methodologies is shown in Table 7.6. Of course, it is identical for all three of them. It comprises of points A, B and C shown in Figure 7.2. Points D and E are also initially identified by the Double-step solution algorithm, but are subsequently eliminated since they are dominated. On the other hand, the other two methodologies do not identify dominated points.

Table 7.6: Non-dominated set of the case study

CFA_h	V_h
7928	9211.556
7628	8444.889
7328	6778.222

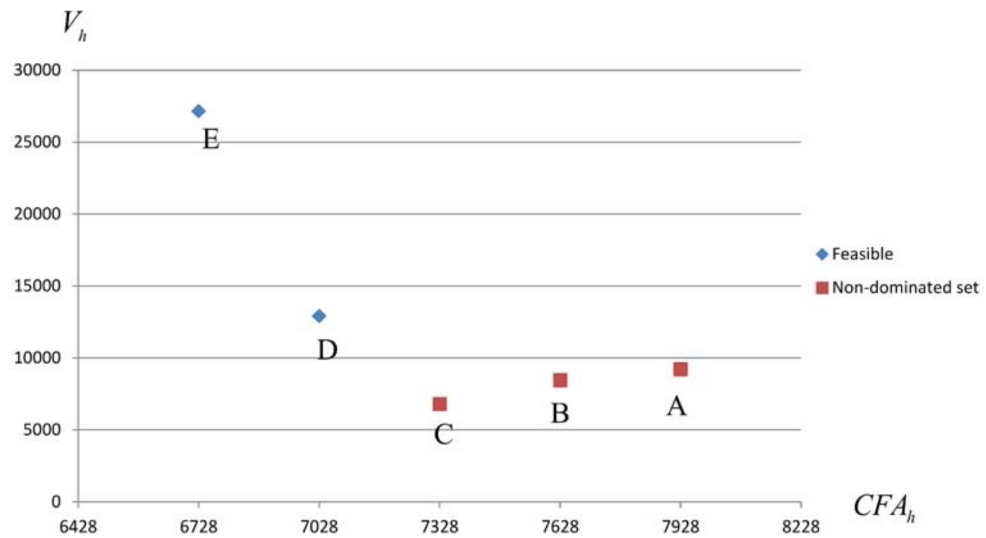

Figure 7.2: Non-dominated set of the case study

7.4 Computational implementation

In this section, we present experimental results evaluating the computational performance of the proposed solution algorithms and demonstrating their efficiency. All three algorithms were implemented in C/C++ interfacing with LINGO 13.0 through LINGO Dynamic Link Library (DLL) callback functions. LINGO 13.0 (2011) is a commercial optimization software package that can accommodate the mixed integer quadratic formulations involved. In order to demonstrate the applicability of the proposed methodologies and highlight the benefits that can result from their actual implementation, we test them on random problem instances whose size and characteristics resemble those of realistic problem instances encountered in the typical operation of the HAF. Our computational experiments were performed on an i5-3330 Intel Quad Core processor @ 3.0 GHz with 16 GB system memory.

Due to the fact that the complexity of the single objective models AUGMECON2 is faced with makes the treatment of large size problem instances such as those encountered in realistic environments impracticable, we chose five small values (i.e., 10, 15, 20, 25 and 30) for the total number of aircraft comprising the unit. On the other hand, since they are treating suitable relaxations of the original model, the other two proposed methodologies are capable of handling considerably larger problem instances; therefore, we also tested their performance on larger problem instances comprising of 50 and 100 aircraft. 30 random problem instances were solved for each size considered. The size of the planning horizon was always set equal to six monthly time periods, motivated by the fact that the unit command typically issues the flight load requirements over a six-month planning horizon. LINGO was mainly invoked with default options and the Global Solver enabled to ensure that the global optimal solution was always obtained.

The random problem generator was specially designed to generate problems which resemble the realistic ones as closely as possible. The specifics are as follows: In practice, C is equal to roughly $0.1N$ (for a unit comprising of 60-100 aircraft, the maintenance facility can typically accommodate 6-10 aircraft). Since increasing the value of C appears to increase the computational burden, we set C equal to $0.15N$ rounded down to the next integer in order to make the posed problem instances more challenging. The number of aircraft which were initially grounded, NA, was determined randomly, using a discrete probability function with possible values the integers between 0 and C, inclusive. Negative skewness was imposed on this distribution, resulting in higher probabilities for larger NA values. More specifically, the probability that NA was equal to x, for $x = 0,\ldots,C$, was set equal to $(x+1)/(\sum_{x=0}^{C}(x+1))$. Naturally, the number of aircraft which were initially available, A, was set equal to N-NA.

For each available aircraft, its residual flight time was uniformly distributed in the interval $[Y_{min},Y]$, while, for each grounded aircraft, its residual maintenance time was uniformly distributed in the interval $[G_{min},G]$. The flight load requirement of each time period was a random number uniformly distributed in the interval $[10N, 15N]$, while the time capacity of the maintenance facility in each time period was a random number uniformly distributed in the interval $[15N, 20N]$. This random generation scheme results in problem instances with characteristics resembling closely those of realistic problems, with the specifics reasons for this being strictly confidential. For the remaining problem parameters, we chose realistic values, i.e., $Y = 300$, $Y_{min} = 0.1$, $G = 320$, $G_{min} = 0.1$, and $X_{max} = 50$.

Table 7.7 presents the average and maximum computational times of the three solution algorithms, i.e., AUGMECON2, the Double-step solution algorithm, and the Hybrid approach. Two variants of the Double-step algorithm are considered, based on whether a feasibility check is performed in advance for each CFA_h level as described in 7.3.3.2 (FEAS) or not. The entries for $N >$ 15 are missing for AUGMECON2, because the optimization solver was not able to accommodate such instances within the 12-hour limit that was enforced. Additionally, 11 problem instances with $N = 15$ did not terminate within this time limit; therefore, the average and maximum computational time for N = 15 has been computed over 19 instances instead of 30.

Table 7.7: Computational times (in seconds) of the three methodologies (CFA_h objective)

	AUGMECON2 (original model)		**Double-step** (relaxed model)		**Double-step (FEAS)** (relaxed model)		**Hybrid** (relaxed model)	
N	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**
10	195.86	1010.21	1.73	3.06	2.22	3.45	1.06	1.69
15	1410.52	6883.39	9.39	19.11	11.65	24.13	13.49	67.00
20			34.43	95.19	37.97	89.11	27.67	142.85
25			94.73	330.19	115.95	569.13	36.62	116.23
30			101.04	493.98	107.23	493.16	82.90	566.56
50			888.82	4808.69	957.45	6453.72	341.34	7321.84
100			1302.09	4987.47	1306.09	5104.39	602.17	8019.53

The superiority of the two proposed solution methodologies becomes immediately evident, since their computational times are significantly lower than those exhibited when AUGMECON2 is applied to the original model. Of course, this behavior should not be considered as an AUGMECON2 deficiency, but should be attributed to the solution complexity of the optimization models the method is faced with. Since the solution of these models appears impracticable even for moderate size problem instances, the computational savings realized when the other two methodologies are utilized are excessive, as the results of Table 7.7 demonstrate. The computational performance of the Hybrid solution methodology appears superior to that of the Double-step solution methodology as the problem size increases. The variability of the computational times of all three solution methods appears significant. This is further supported by the fact that, in a few cases, increasing the problem size results in a reduction of the computational times. Performing the feasibility check in advance does not seem to improve the computational performance of the Double-step solution algorithm. Nevertheless, the employment of this step is motivated by its ability to mitigate the potential risk of extensive execution times due to the existence of an exponential number of intermediate infeasible solutions.

Table 7.8 presents results regarding the cardinality of the non-dominated set, and the number of LINGO calls, i.e., the number of optimization problems each methodology solves in order to find the non-dominated set. As these results demonstrate, the size of the non-dominated set appears to increase moderately with the size of the underlying problem. On the other hand, for the same problem size, the fewest optimization problems are solved by AUGMECON2. The considerably larger computational times of this methodology lead to the conclusion that the solution of these problems is substantially more time consuming than that of the problems solved by the other methodologies. As far as the other two methodologies are concerned, the Hybrid algorithm solves significantly fewer optimization problems than the Double-step algorithm. In addition, performing the feasibility check in advance does not appear to benefit the Double-step algorithm, leading to an increased number of optimization problems solved.

Table 7.8: N_d cardinality and number of LINGO calls for the three methodologies (CFA_h objective)

	N_d set cardinality		AUGMECON2 (original model)		Double-step (relaxed model)		Double-step (FEAS) (relaxed model)		Hybrid (relaxed model)	
N	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
10	1.53	3	4.82	6	14.57	20	19.86	27	6.82	11
15	3.10	7	5.84	10	24.74	34	34.16	45	9.74	16
20	2.70	5			39.07	50	53.43	66	9.60	15
25	4.93	8			49.13	60	68.17	81	14.03	20
30	5.47	9			61.27	70	84.23	95	15.20	23
50	9.20	15			110.48	127	150.93	173	22.86	34
100	17.33	28			213.57	257	276.48	327	40.35	60

7.5 Algorithmic enhancements & extensions

In this section, we discuss some algorithmic enhancements and potential model extensions. In particular, we elaborate on alternative model objectives, we discuss the potential parallelization of the algorithmic implementations in order to take advantage of modern computer architecture, and we study a 3-objective model extension.

7.5.1 Alternative objectives

As briefly discussed in Section 3.5, in the military context, the readiness of an aircraft wing is typically determined in terms of the total number of available aircraft (aircraft availability) and in terms of the total residual flight time (residual flight time availability). Mathematically, the cumulative aircraft availability objective is expressed as follows:

$$\text{Max } CFA_a = \text{Max}\sum_{t=2}^{T+1}\sum_{n=1}^{|N|} a_{n,t}$$

The corresponding biobjective formulation (Bi-FMPa) that incorporates the minimization of the availability variability is the following in this case:

$$\text{Max } CFA_a(x)$$
$$\text{Min } V_a(x) \qquad \text{(Bi-FMP}_\text{a}\text{)}$$
$$\text{s.t. } x \in X \text{ ,}$$

where $V_a(x) = \left(\frac{1}{T}\right)\sum_{t=2}^{T+1}\left(SA_t(x) - \overline{SA}(x)\right)^2$, $SA_t = \sum_{n=1}^{|N|} a_{n,t}$, and $\overline{SA}(x) = \frac{CFA_a(x)}{T} = \frac{\sum_{t=2}^{T+1} SA_t(x)}{T}$.

The design of the proposed solution algorithms enables them to cater for the aircraft fleet availability objective, too. A slight modification deemed necessary in this case is the fact that the grid points used for the CFA_a objective must differ by a step size of 1 instead of Y. Moreover, a valid upper bound on the ideal value of the aircraft fleet availability is clearly $|N|T$, whereas a lower bound on its nadir value is clearly $(|N|-C)T$. Table 7.9 presents the average and maximum computational times of the three solution algorithms for Problem Bi-FMP$_a$. The same two variants are again considered for the Double-step solution algorithm.

Table 7.9: Computational times (in seconds) of the three methods (CFA_a objective)

	AUGMECON2 (original model)		**Double-step** (relaxed model)		**Double-step (FEAS)** (relaxed model)		**Hybrid** (relaxed model)	
N	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**	**Avg**	**Max**
10	65.03	285.31	0.92	1.69	1.34	2.59	1.38	2.70
15	913.53	1149.52	4.12	10.61	5.24	13.47	6.68	18.53
20	2446.26	9034.04	13.71	40.68	14.63	45.15	20.61	95.14
25			19.57	45.49	23.79	105.06	24.63	52.62
30			26.90	108.45	47.47	182.41	31.82	145.74
50			201.90	692	209.55	720.03	314.08	1632.30
100			1882.32	6117.53	1690.98	4209.70	2600.26	8003.20

Compared to the instances in which the residual flight time availability is subject to optimization, the computational times of the two proposed solution algorithms appear higher for some problem sizes, and lower for some other ones. The computational performance of AUGMECON2 appears improved, since the maximum problem size for which it successfully computes the non-dominated set is slightly larger than before. The superiority of the two proposed solution methods over AUGMECON2 is evident in this case, too and can be attributed to the same reasons as before. In contrast to Bi-FMP$_h$, the computational performance of the Double-step solution algorithm appears superior in this case. Moreover, with the exception of the instances with N = 100, performing the feasibility check in advance still has a slightly negative effect.

Table 7.10 presents similar results as those of Table 7.8 for the CFA_a objective. As these results demonstrate, the size of the non-dominated set seems to slightly increase when the CFA_a objective is adopted instead of the CFA_h. On the other hand, the effect on the number of optimization problems solved appears mixed, as it is higher in some cases, and lower in some other ones. AUGMECON2 still solves the fewest optimization problems among the three methods, while the Hybrid algorithm still solves fewer optimization problems than the Double-step algorithm. Finally, performing the feasibility check in advance still leads to an increased number of optimization problems solved for the Double-step algorithm.

Table 7.10: N_d cardinality and number of LINGO calls for the three methodologies (CFA_a objective)

	N_d set cardinality		AUGMECON2 (original model)		Double-step (relaxed model)		Double-step (FEAS) (relaxed model)		Hybrid (relaxed model)	
N	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max
10	2.33	4	3.73	8	11.50	12	17.50	18	8.60	12
15	4.30	7	6.28	12	21.07	22	32.07	33	12.63	18
20	6.53	12	8.41	18	30.29	32	45.96	48	17.57	28
25	7.40	12			35.41	37	50.41	55	19.07	32
30	6.87	17			41.45	42	62.45	63	21.59	38
50	16.10	33			71.18	72	107.18	108	47.41	70
100	30.93	42			145.83	152	212.33	228	39.06	88

7.5.2 Parallelization

The inherent structure of the proposed solution methodologies makes possible the execution of several algorithmic tasks in parallel, taking advantage of modern computer architecture, and in particular multi-core processors. More precisely, once the CFA_h search range has been identified, the augmented ε-constraint model can be solved independently for each distinct grid point ε_k at a separate CPU thread. After all grid points have been considered, the non-dominated points together with their corresponding Pareto optimal solutions can be identified.

The parallelization is possible, because the solution of the problem for the different grid points can be carried out independently of each other, likewise the AUGMECON2 method (Florios and Mavrotas, 2014). Our computational experience suggests that the main computational burden of the algorithm lies in the execution of this task; therefore, the proposed parallelization is expected to provide a substantial speedup.

7.5.3 A 3-objective model extension

In order to investigate the proposed model's behavior under the presence of more than two objectives, we consider a 3-objective model extension in this subsection, and we outline how the proposed methodology can be modified in order to accommodate it. We also study the effect of the inclusion of the third objective on the cardinality of the non-dominated set, as well as on the algorithmic computational performance. The proposed approach for accommodating the 3-objective FMP model is illustrated on a small case study which comprises a suitable extension of the case study introduced in Section 7.3.5.

Whereas maximizing the unit's cumulative fleet availability over the entire planning horizon is one of the most important objectives of the FMP problem, ensuring that the fleet availability will not fall below a critical level in any time period of the planning horizon is also of major importance. With this in mind, we incorporate the additional objective of maximizing the minimum fleet availability ensured for each time period of the planning horizon into the proposed model. Mathematically, this objective is expressed as follows:

$$\text{Max } SY_{min}$$
$$\text{s.t. } SY_{min} \leq SY_t,\ t = 2,\ldots,T+1$$

In turn, this leads to the following multiobjective formulation for the FMP problem (Multi-FMP_h):

$$\begin{aligned} &\text{Max } CFA_h(x) \\ &\text{Min } V_h(x) \\ &\text{Max } SY_{min}(x) \\ \text{s.t. } & SY_{min} \leq SY_t = \sum_{n=1}^{|N|} y_{n,t},\ t = 2,...,T+1, \\ & x \in X \end{aligned} \qquad \text{(Multi-FMP}_h\text{)}$$

Based on the key result that the domain comprising possible CFA_h values is a discrete set, we modify accordingly the Double-step algorithm next, enabling the treatment of this 3-objective FMP problem formulation, too. In Section 7.3.3, it was demonstrated that the Double-step algorithm successfully obtains the entire non-dominating set of Problem Bi-FMP_h by fixing the CFA_h objective in a stepwise manner and minimizing for each of these values the associated variability, $V_h(x)$. We extend this approach, by applying a suitable modification of the Double-step solution algorithm for each of these distinct CFA_h values. This involves fixing the CFA_h level in successive iterations, and computing the associated non-dominated set with respect to the other two objectives by increasing the SY_{min} objective in a stepwise manner and computing the minimum variability associated solution. After all possible CFA_h levels have been considered, a simple procedure eliminates out of all the solutions that have been identified those which are dominated with respect to all 3 objectives. Ensuring that both the CFA_h objective (at the outer level), as well as the SY_{min} objective (at the inner level) are increased in a stepwise fashion ensures that no non-dominated solution will be overlooked; thus, the entire non-dominated set will be obtained upon termination.

For $t = 2,\ldots,T+1$, the residual flight time availability of the unit at the beginning of time period t, SY_t, is equal to $\sum_{n=1}^{|N|} Y1_n - \sum_{k=1}^{t-1} S_k + Y \cdot \sum_{k=2}^{t} (ex_k)$. Moreover, for $t = 2,\ldots,T+1$, the nominal combination imposes upper bounds on the cumulative number of aircraft exiting the maintenance facility from the beginning of the planning horizon up until time period *t*; thus, $0 \le \sum_{k=2}^{t} ex_k \le \sum_{k=2}^{t} ex_k^{\text{nom}}$ holds true in any feasible solution. Hence, we can compute all possible values of the unit fleet availability at the beginning of time period *t*, for *t* = 2,…,*T*+1, by considering all feasible values of the summation $\sum_{k=2}^{t} ex_k$ imposed by this constraint. These values can then be used to define SY_{min} objective steps in the Double-step instances employed for the various CFA_h levels. Next, we illustrate how this can be accomplished for the small case study introduced in Section 7.3.5.

For *t*=2, $ex_2^{\text{nom}} = 0$ according to Table 7.5. Subsequently, the only possible value for SY_2 is $\sum_{n=1}^{|N|} Y1_n - S_1 + Y * 0 = 1321 - 144 + 300 * 0 = 1177$. Calculating all possible values for the remaining 5 time periods, we conclude that the domain comprising possible SY_{min} values is the following set: {554, 685, 813, 854, 944, 985, 1055, 1113, 1154, 1177, 1244, 1285, 1355, 1413, 1454}. For each of the three possible values of the CFA_h objective, we employ a distinct Double-step solution algorithm instance, which computes the non-dominated set with respect to the V_h and SY_{min} objectives. At the end, we perform a final step that eliminates out of all identified solutions the dominated ones with respect to all 3 objectives, and we conclude that the non-dominated set is the one presented in Table 7.11 and depicted in Figure 7.3.

As far as the computational burden is concerned, the number of LINGO iterations increases from 15 in the 2-objective case to 170 in the 3-objective case, although many LINGO calls turn out to be infeasible/redundant. Nevertheless, the total execution time increases by nearly 8 times. Of course, a more sophisticated solution approach fully exploiting the problem's theoretical properties will eliminate many redundant iterations, improving the computational time substantially. Even in that case, however, the computational requirements are expected to be considerably larger and increase more progressively with problem size. Our elementary computational experience suggests that the application of the proposed simplistic solution procedure remains practical only for problems with up to 30-40 aircraft. Considering the fact that realistic FMP instances include up to 100 aircraft, it becomes apparent that the development of a more sophisticated solution algorithm for the 3-objective case of the problem turns up as a very promising direction for future research.

Table 7.11: Non-dominated set of the multiobjective case study

CFA_h	V_h	SY_{min}
7928	9211.556	1177
7628	8444.889	1154
7328	6778.222	1113

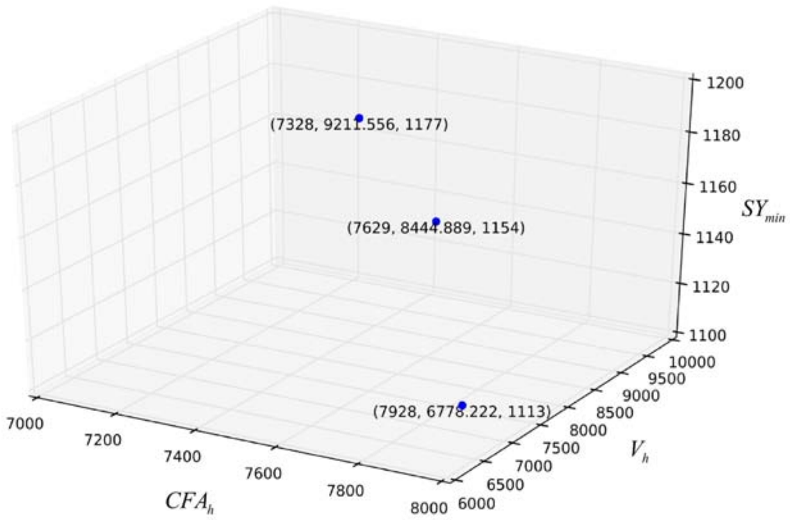

Figure 7.3: Non-dominated set of the multiobjective case study

7.6 Summary

In this chapter, we addressed a biobjective quadratic model for the FMP problem. The problem consists of compiling individual flight and maintenance plans for a group of aircraft that comprise an aircraft wing. The aim is to maximize the fleet availability of the unit, while also minimizing its variability. For this problem, we developed two exact solution methodologies which are capable of identifying the entire non-dominated set. The first one minimizes the variability of the fleet availability, while also decreasing the fleet availability level in a stepwise fashion; this ensures that no non-dominated solutions are overlooked. The second one is a hybrid approach, which combines the computational savings gained from the introduction of the payoff table calculation through lexicographic optimization and slack variables, with those gained from the disaggregation of the FMP solution into several steps. The performance of the two proposed algorithms on problems with realistic characteristics appears to be considerably superior to that of the traditional ε-constraint approach, which can be used alternatively for the solution of this problem. The main reason for this seems to be the considerably high computational requirements of the optimization models the ε-constraint approach encounters.

Chapter 8 Dissertation Summary and Concluding Remarks

As stated in Chapter 1, the main contribution of the present work lies in the development of various mathematical optimization models for the formulation of the FMP problem, along with the specialized algorithms that facilitate their efficient solution. In this chapter, we conclude this dissertation by presenting the most prominent findings and results of this research. First we provide a review of the dissertation, then we make some suggestions for facilitating the application of the proposed models, and lastly we provide some promising directions for future research.

Chapter 1 is an introductory chapter, giving an overview of the motivation for dealing with the specific problem under consideration, stating the dissertation's main contributions, and providing an outline of the following chapters. In Chapter 2, we review the related literature, focusing mostly on works that address military related applications. In Chapter 3, we present a detailed definition of the FMP problem, and we address model formulation considerations related to the development of accurate FMP optimization models.

In Chapter 4 we prove several interesting theoretical properties of the FMP problem, and we utilize them to develop two heuristic solution approaches for solving large FMP instances. We also present experimental results demonstrating the computational performance of these heuristics and the quality of the solutions they produce. The first heuristic, AFH, exhibits a very satisfactory performance in most of the cases, which justifies its wide usage by many Air Force organizations worldwide. The second heuristic, HSH, exhibits a rather myopic behavior. It works by splitting the original planning horizon into smaller ones, and solving an FMP problem for each of them. Although this technique may result in low availability over the last sub-horizons, the solution obtained by HSH is also quite satisfactory in most of the cases. Therefore, it can be considered as an alternative choice for identifying a solution of satisfactory quality when the size of the problem prohibits the application of an exact solution methodology.

In Chapter 5, we develop a mixed integer nonlinear model for the FMP problem, which is based on a suitable modification of an existing graphical heuristic tool for addressing this problem. Utilizing the problem's special structure and theoretical properties, we also develop an exact solution algorithm for accommodating this model. Our computational results demonstrate that the performance of the proposed solution algorithm is superior compared to that of a commercial optimization package.

In Chapter 6, we consider a mixed integer optimization model for the multi-period version of the FMP problem. For this model, we develop an exact solution algorithm that identifies a valid upper bound on the optimal objective first, and then reduces this bound in a stepwise fashion until a feasible solution that attains it is identified. The performance of the algorithm on realistic problem instances appears superior to that of two commercial optimization solvers that can be used alternatively for the solution of the problem, whereas the opposite behavior is observed for a class of problems with significantly different characteristics.

In Chapter 7, we address a biobjective quadratic FMP model, which incorporates the minimization of the fleet availability variability. For this FMP problem variant, we develop two exact solution methodologies which are capable of obtaining the entire set of non-dominated solutions. The first methodology disaggregates the original FMP model into smaller subproblems whose solution is attained much more efficiently. The second methodology is a variant of the ε-constraint method, applied to a suitable relaxation instead of the original FMP model. The performance of the two proposed methodologies on problems with realistic characteristics appears to be considerably superior to that of the traditional ε-constraint approach, which can be used alternatively for the solution of the problem.

The present work provides an in-depth study of several interesting variants of the FMP problem. The most important contribution of this work is the development of several interesting optimization models for this problem, along with the specialized algorithms that facilitate their efficient solution, which comprise an efficient toolset the aviation/maintenance managers can utilize to address the numerous aspects of the FMP problem effectively. The extensive computational results that we present demonstrate the performance of the proposed solution methodologies.

The key objective of the FMP problem is to maximize the *operational readiness* of a military unit, as stated in Chapter 1, mainly through the maximization of the fleet availability. The developed models can provide valuable information to the aviation/maintenance managers in many ways. For example, by using current data as input to these models, they can assess the midterm operational readiness of the unit. Moreover, the aviation/maintenance managers can utilize the proposed models to perform what-if scenarios and test alternative parameter choices and desing options. For example, if the fleet availability that results with the existing maintenance resources turns out to be insufficient, the aviation/maintenance managers can relax the maintenance costraints in order to find out the extra maintenance resources needed to achieve the desired availability level, and then acquire the extra personell. Another valuable use of the proposed models is for examining how new operational requirements, such as exercises and deployments, can influence the long-term readiness of the fleet. By relaxing the maintenance costraints, the aviation/maintenance managers can estimate the extra maintenance capacity needed to handle the additional operational load. The fact that they enable the examination of such hypothetical scenarios gives added value to the developed mathematical models, fulfilling the original research aims.

The present work points to several promising directions for future research. The deterministic models that we address in this work comprise a basic building block towards developing more complicated models that will take into consideration stochastic events, such as unforeseen failures.

Future works should be directed towards the development of such stochastic models that will incorporate the uncertainty that some of the problem's parameters might exhibit. A recent work in that direction is the paper by Mattila and Virtanen (2014), who use discrete event simulation to model the maintenance of military aircraft in the Finnish Air Force and study its impact on aircraft availability. We deliberately decided to deal with the deterministic version of the FMP problem, because it is novel, complex, interesting and realistic enough to constitute a contribution in itself, which is useful for researchers and practitioners.

Two factors of the problem under consideration exhibit significant uncertainty in practice: the fact that the actual duration of the maintenance service may turn out to be longer than its nominal value (expressed by parameter G), and the fact that a mission aircraft may fail in fulfilling its entire flight load (expressed by decision variable x_i). The grounding of an aircraft for a longer time period than the one determined by parameter G may be dictated due to the detection of serious findings during the regularly scheduled maintenance inspection, or due to the lack of specific spare parts or staff expertise, which are needed in order to complete the service of the aircraft according to the prescribed safety standards. On the other hand, an aircraft may not be able to fly the entire time that has been assigned to it, due to an unexpected event such as an unforeseen failure. Of course, in case of such undesirable events, the user always has the option of reapplying the proposed model after updating the state of the system, but a stochastic model will clearly incorporate such uncertainties more accurately, leading to a better long term performance for the entire system.

Another interesting direction for future research appears to be the suitable modification of the model studied in Chapter 5 that will enable its application to a multi-period planning horizon, so that its long term performance and behavior can be evaluated. This will also render this model directly comparable to other models that have been proposed in the related literature, which are targeted towards multi-period planning horizons. Additionally, for the biobjective quadratic version of the FMP problem studied in Chapter 7, the generation of the exact non-dominated set opens up the opportunity of applying state-of the-art multi-objective metaheuristics and using the benchmarks provided in order to assess their effectiveness. Finally, the study of FMP multi-objective models with more than two objectives such as the one introduced in Section 7.5.3 also stems as a very interesting direction for future research.

Appendix A Proof of Proposition 4.1

Let λ_1, λ_2 and u_i (i = 1,…, $|K|$) be the non-negative dual multipliers of constraints (4.3.2), (4.3.3) and (4.3.4), respectively. In addition to the original constraints of the problem, the KKT conditions are:

$$-2(y_{kp} - ks - x_k) - \lambda_1 + \lambda_2 + u_k \geq 0,\ k = 1,...,|K| \tag{1}$$

$$x_k[-2(y_{kp} - ks - x_k) - \lambda_1 + \lambda_2 + u_k] = 0,\ k = 1,...,|K| \tag{2}$$

$$\lambda_1(LS - \sum_{m \in M_1} y_{mp} - \sum_{k=1}^{|K|} x_k) = 0 \tag{3}$$

$$\lambda_2(\sum_{k=1}^{|K|} x_k - US + \sum_{m \in M_1} y_{mp}) = 0 \tag{4}$$

$$u_k(x_k - X_k) = 0,\ k = 1,...,|K| \tag{5}$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, u_k \geq 0, k = 1,\ldots, |K| \tag{6}$$

We call the quantity ($y_{kp} - ks - x_k$) the "perpendicular distance of aircraft k from the diagonal". Despite the word "distance" in this definition, note that this quantity is negative when aircraft k lies below the diagonal at the end of the current period. Since $LL \leq UL$ always, there are 12 possible distinct arrangements of the quantities LL, UL, D and X. When there does not exist an arrangement in which LL precedes X, the problem is clearly infeasible, since the flight requirements (constraint (4.3.2)) cannot be satisfied, even when every aircraft is assigned its maximum possible flight time. In each of the remaining 8 cases, it is clear that the solution obtained from the application of the Procedure Sweep when the sum of the assigned aircraft flight times becomes equal to the second quantity in the arrangement, satisfies (4.3.2)-(4.3.4) and is therefore feasible. We show next that this solution also satisfies conditions (1)-(6), and is therefore optimal, too.

Case 1: The arrangement is {LL,UL,D,X} or {LL,UL,X,D}.

In this case, the sum of the assigned aircraft flight times in the obtained solution is equal to the second quantity in the arrangement, UL. We partition the indices of the decision variables of this solution into 4 sets:

a) Set S_1 contains the indices of the variables x_k such that $x_k = 0 = X_k$.

b) Set S_2 contains the indices of the variables x_k such that $x_k = 0 < X_k$,

c) Set S_3 contains the indices of the variables x_k such that $0 < x_k = X_k$,

d) Set S_4 contains the indices of the variables x_k such that $0 < x_k < X_k$.

We set λ_1 = 0 and $\lambda_2 = 2(y_{kp} - ks - x_k)$ for some $k \in S_4$. Note that the value of λ_2 is the same for any $k \in S_4$, since set S_4 contains the indices of the variables that lie on the sweeping line at the current solution, and, as a result, their perpendicular distance from the diagonal is the same. Additionally, each of these distances is non-negative, since the fact that *UL* appears before *D* in the arrangement implies that the sweeping line lies above the diagonal at the current solution; therefore λ_2 is non-negative, too. We also set $u_k = \max(2(y_{kp} - ks - x_k) - \lambda_2, 0)$ for $k \in S_1$, $u_k = 0$ for $k \in S_2 \cup S_4$ and $u_k = 2(y_{kp} - ks - x_k) - \lambda_2$ for $k \in S_3$. The multipliers u_k for $k \in S_3$ are always non-negative, since set S_3 contains the indices of the variables that were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, the perpendicular distance of each of these points from the diagonal cannot be smaller than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. For $k \in S_1$ and $k \in S_3$, constraints (1) and (2) are clearly satisfied. Constraints (1) are clearly satisfied as an equality for $k \in S_4$; therefore, constraints (2) are satisfied, too. For $k \in S_2$, constraints (2) are clearly satisfied and constraints (1) are satisfied if $\lambda_2 \geq 2(y_{kp} - ks - x_k)$, which is true, since set S_2 contains the indices of the variables that have not been swept by the line yet; therefore, their perpendicular distance from the diagonal cannot be larger than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. Finally, constraints (3)-(6) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_k ($k = 1,\ldots,|K|$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

If set S_4 is empty, then the above analysis remains the same, but λ_2 needs to be set equal to $\max(\max_{k \in S_2} 2(y_{kp} - ks - x_k), 0)$. If both S_2 and S_4 are empty, then every decision variable has taken its maximum possible value. This implies that $UL = X$, and this case reduces to Case 4 through an appropriate rearrangement.

Case 2: The arrangement is {*LL*,*D*,*UL*,*X*} or {*LL*,*D*,*X*,*UL*}.

In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, *D*, which implies that the sweeping line coincides with the diagonal. We partition the indices of the decision variables of this solution into the same 4 sets as in Case 1. We set $\lambda_1 = \lambda_2 = 0$, $u_k = \max(2(y_{kp} - ks - x_k), 0)$ for $k \in S_1$, $u_k = 0$ for $k \in S_2 \cup S_4$ and $u_k = 2(y_{kp} - ks - x_k)$, for $k \in S_3$. The multipliers u_k for $k \in S_3$ are always non-negative, since set S_3 contains the indices of the variables that were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, since the sweeping line coincides with the diagonal at the current solution, the perpendicular distance of each of these points from the diagonal cannot be negative. For $k \in S_1$ and $k \in S_3$, constraints (1) and (2) are clearly satisfied. For $k \in S_2$, constraints (2) are clearly satisfied and constraints (1) are satisfied if $-2(y_{kp} - ks - x_k) \geq 0$, which is true, since set S_2 contains the indices of the variables that have not been swept by the line yet; therefore, since the sweeping line coincides with the diagonal at the current solution, each of these points has non-positive perpendicular distance from the diagonal. Set S_4 contains the indices of the variables that lie on the sweeping line at the current solution. Since the sweeping line coincides with the diagonal, the perpendicular distance of each of these variables from the diagonal is equal to 0. As a result, constraints (1) and (2) are also satisfied for $k \in S_4$. Finally, constraints (3)-(6) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_k (k = 1,…,$|K|$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

Case 3: The arrangement is $\{D,LL,UL,X\}$ or $\{D,LL,X,UL\}$.

In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, LL. We partition the indices of the decision variables of this solution into the same 4 sets as in Cases 1 and 2. We set $\lambda_2 = 0$ and $\lambda_1 = -2(y_{kp} - ks - x_k)$, for $k \in S_4$. Note that the value of λ_1 is the same for any $k \in S_4$, since set S_4 contains the indices of the variables that lie on the sweeping line at the current solution, and, as a result, their perpendicular distance from the diagonal is the same. Additionally, each of these distances is non-positive, since the fact that D appears first in the arrangement implies that the sweeping line does not lie above the diagonal at the current solution; therefore, λ_1 is non-negative, too. We also set $u_k = \max(2(y_{kp} - ks - x_k) + \lambda_1, 0)$ for $k \in S_1$, $u_k = 0$ for $k \in S_2 \cup S_4$ and $u_k = 2(y_{kp} - ks - x_k) + \lambda_1$ for $k \in S_3$. The multipliers u_k for $k \in S_3$ are always non-negative, since set S_3 contains the indices of the variables that were initially swept and later disengaged by the sweeping line because they reached their upper bound; therefore, the perpendicular distance of each of these points from the diagonal cannot be smaller than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. For $k \in S_1$ and $k \in S_3$, constraints (1) and (2) are clearly satisfied. Constraints (1) are clearly satisfied as an equality for $k \in S_4$; therefore, constraints (2) are satisfied, too. For $k \in S_2$, constraints (2) are clearly satisfied and constraints (1) are satisfied if $-\lambda_1 \geq 2(y_{kp} - ks - x_k)$, which is true, since set S_2 contains the indices of the variables that have not been swept by the line yet; therefore, the perpendicular distance of each of these points from the diagonal cannot be larger than the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution. Finally, constraints (3)-(6) are clearly satisfied, too. Hence, the current solution together with λ_1, λ_2 and u_k (k = 1,…,$|K|$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

If set S_4 is empty, then the above analysis remains the same, but λ_1 needs to be set equal to $\min_{k \in S_2}(-2(y_{kp} - ks - x_k))$, which is always non-negative, since set S_2 contains the indices of the variables that have not been swept by the line yet; therefore, since the sweeping line does not lie above the diagonal at the current solution, each of these points has non-positive perpendicular distance from the diagonal. If both S_2 and S_4 are empty, then every decision variable has taken its maximum possible value. If $D = LL = X$, this case reduces to Case 4 through an appropriate rearrangement. If $D < LL = X$, then the sweeping line lies below the diagonal at the current solution. In this case, we partition the indices of the decision variables of the current solution into 2 sets:

a) Set S_1 contains the indices of the variables x_k such that $x_k = 0 = X_k$,

b) Set S_2 contains the indices of the variables x_k such that $0 < x_k = X_k$.

We set $\lambda_1 = -2(y_{kp} - ks - x_k)$, where $k \in S_2$ is the index of a variable that is currently on the sweeping line, $\lambda_2 = 0$, $u_k = \max(2(y_{kp} - ks - x_k) + \lambda_1, 0)$ for $k \in S_1$ and $u_k = 2(y_{kp} - ks - x_k) + \lambda_1$, for $k \in S_2$. Since the sweeping line lies below the diagonal, λ_1 is strictly positive. Additionally, the multipliers u_k for $k \in S_2$ are always non-negative, since the perpendicular distance from the diagonal of any point that lies on the sweeping line at the current solution cannot be larger than the perpendicular distance from the diagonal of any other point. Constraints (1)-(6) are clearly satisfied for $k \in S_1 \cup S_2$. Hence, the current solution together with λ_1, λ_2 and u_k ($k = 1,\ldots,|K|$) as the dual multipliers satisfies the KKT conditions and is therefore optimal.

Case 4: The arrangement is $\{LL,X,UL,D\}$ or $\{LL,X,D,UL\}$.

In this case, the sum of the assigned aircraft flight times in the solution obtained is equal to the second quantity in the arrangement, X. We partition the indices of the decision variables of this solution into 2 sets:

a) Set S_1 contains the indices of the variables x_k such that $x_k = 0 = X_k$,

b) Set S_2 contains the indices of the variables x_k such that $0 < x_k = X_k$.

We set $\lambda_1 = \lambda_2 = 0$, $u_k = \max(2(y_{kp} - ks - x_k), 0)$ for $k \in S_1$ and $u_k = 2(y_{kp} - ks - x_k)$ for $k \in S_2$. The multipliers u_k for $k \in S_2$ are always non-negative, since set S_2 contains the indices of the variables that have already been swept by the sweeping line to their upper bound; therefore, since the sweeping line does not lie below the diagonal at the current solution, their perpendicular distance from the diagonal is non-negative. Constraints (1)-(6) are clearly satisfied for $k \in S_1 \cup S_2$. Hence, the current solution, together with λ_1, λ_2 and u_k ($k = 1,\ldots,|K|$) as the dual multipliers, satisfies the KKT conditions and is therefore optimal. □

Appendix B Proof of Lemma 4.1

The values of *LL* and *UL* are known. We can compute the values of *D* and *X* in time O(|*K*|), since $D = \sum_{k=1}^{|K|} \min(X_k, [y_{kp} - ks]^+)$ and $X = \sum_{k=1}^{|K|} X_k$. Finding if there exists an arrangement of *LL*, *UL*, *D* and *X* in which *LL* precedes *X* requires time O(1). If such an arrangement exists and *Q* is the second quantity in order, we can equivalently transform the problem defined by (4.3.1)-(4.3.4) into the following problem in time O(|*K*|):

$$\operatorname*{Min}_{x_k} \sum_{k=1}^{|K|} \left(\frac{1}{2} d_k x_k^{\,2} - a_k x_k\right)$$
$$\text{s.t.} \quad \sum_{k=1}^{|K|} b_k x_k = b_0$$
$$l_k \le x_k \le u_k,\ k = 1,\ldots,|K|,$$

where $d_k = 2$, $a_k = 2(y_{kp} - ks)$, $b_k = 1$, $b_0 = Q$, $l_k = 0$ and $u_k = X_k$, for $k = 1,\ldots,|K|$. We have suppressed the term $(y_{kp} - ks)^2$ in this formulation, since it is constant and does not affect the optimization. This problem can be solved in time O(|*K*|) (see Brucker (1984)). Therefore, the problem defined by (4.3.1)-(4.3.4) can be solved in total time O(|*K*|), too. □

Appendix C List of Dissertation Publications

Parts of the work presented in this dissertation have been published in scientific journals and presented in international conferences as follows:

Journal Papers

[J.1] Kozanidis, G., Gavranis, A., & Kostarelou, E. (2012). Mixed integer least squares optimization for flight and maintenance planning of mission aircraft. *Naval Research Logistics*, *59*(3-4), 212–229. http://doi.org/10.1002/nav.21483

[J.2] Kozanidis, G., Gavranis, A., & Liberopoulos, G. (2013). Heuristics for flight and maintenance planning of mission aircraft. *Annals of Operations Research*, *221*(1), 211–238. http://doi.org/10.1007/s10479-013-1376-6

[J.3] Gavranis, A., & Kozanidis, G. (2015). An exact solution algorithm for maximizing the fleet availability of a unit of aircraft subject to flight and maintenance requirements. *European Journal of Operational Research*, *242*(2), 631–643. http://doi.org/10.1016/j.ejor.2014.10.016

[J.4] Gavranis, A., & Kozanidis, G. (2017). Mixed integer biobjective quadratic programming for maximum-value minimum-variability fleet availability of a unit of mission aircraft. *Computers & Industrial Engineering*, *110*, 13–29. http://doi.org/10.1016/j.cie.2017.05.010

Papers in International Conferences

[C.1] Kozanidis, G., Gavranis, A., & Liberopoulos, G. (2008). Heuristics for maximizing fleet availability subject to flight and maintenance requirements. In *10th International Conference on Application of Advanced Technologies in Transportation*. Athens, Greece.

[C.2] Gavranis, A., & Kozanidis, G. (2013). An exact solution algorithm for maximizing the fleet availability of an aircraft unit subject to flight and maintenance requirements. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013* (Vol. II, pp. 1036–1041). Hong Kong: Newswood Limited.

Abstracts and Presentations in International Conferences

[P.1] Gavranis, A and Kozanidis, G (2009). "Modeling techniques and solution approaches for maximizing fleet availability of mission aircraft subject to flight and maintenance requirements." 23rd *European Conference on Operational Research*, Bonn, Germany, 5-8 July.

[P.2] Kozanidis, G, Gavranis, A and Kostarelou E. (2010). "Mixed integer least squares optimization for flight and maintenance planning of mission aircraft". 24th *European Conference on Operational Research*, Lisbon. Portugal, 11-14 July.

[P.3] Gavranis, A and Kozanidis, G (2013), "Mixed integer multi-objective optimization for flight and maintenance planning of mission aircraft" in *22nd International Conference on Multiple Criteria Decision Making,* Malaga, Spain, 17-21 June

In Table C.1, we link each of the above works to the corresponding chapter in this dissertation.

Table C.1: List of publications and association to dissertation chapters.

Chapter 4	**Chapter 5**	**Chapter 6**	**Chapter 7**
[J.2] : entire chapter	[J.1] : entire chapter	[J.3] : entire chapter	[J.4] : entire chapter
[C.1]: early work	[P.2]: early work	[C.2]: early work	[P.3]: early work
[P.1] : entire chapter			

Appendix D Glossary of Dissertation Terms and Acronyms

aircraft flowchart	A 2-dimensional graphical tool utilized, in an ad-hoc manner, as a common empirical approach for addressing the FMP problem (see Figure 3.1).
bank time	*residual flight time*
CFA	Cumulative Fleet Availability
diagonal	The line segment that connects the origin with the point with coordinates (A, Y) at the *aircraft flowchart*, where Y is the *phase interval*.
dock space	Unit *maintenance station* space capacity capability
FMP	Flight and maintenance planning
flight load	Suitable flight requirements issued by the unit command at the beginning of each planning horizon in order to retain a high level of unit readiness.
maintenance station	Unit maintenance station for providing service to the aircraft.
operational readiness	“The capability of a unit/formation, ship, weapon system or equipment to perform the missions or functions for which it is organized or designed. May be used in a general sense or to express a level or degree of readiness.” (NATO, 2015)
phase interval	The total flight time of an aircraft between two maintenance inspections.
phased maintenance	Aircraft intermediate level scheduled maintenance conducted at the unit *maintenance station*.
residual flight time	The total remaining time that each individual available aircraft can fly until it has to undergo a maintenance check.
residual maintenance time	The total remaining time that each non-available aircraft needs in order to complete its maintenance check. The residual maintenance time of an aircraft is positive if and only if this aircraft is undergoing a maintenance check, and is therefore not available to fly.
total deviation index	Index equal to the sum of squares of the vertical distances (deviations) of the points mapping the residual flight times of the individual aircraft from their corresponding target values on the *diagonal*.

References

Abrahao, F. T. M., & Gualda, N. D. F. (2006). Fleet Maintenance Scheduling with an Ant Colony System Approach. *Lecture Notes in Computer Science*, *4150*, 412–419.

Athens News Agency. (2006). Daily News Bulletin in English, 06-08-23. Retrieved July 12, 2016, from http://www.hri.org/news/greek/ana/2006/06-08-23.ana.html

Bajestani Aramon, & Beck J.C. (2013). Scheduling a Dynamic Aircraft Repair Shop with Limited Repair Resources. *Journal of Artificial Intelligence Research*, *47*, 35–70. http://doi.org/10.1613/jair.3902

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear programming: theory and algorithms*. Wiley, New York, NY.

Bérubé, J.-F., Gendreau, M., & Potvin, J.-Y. (2009). An exact ε-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits. *European Journal of Operational Research*, *194*(1), 39–50.

Brucker, P. (1984). An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, *3*(3), 163–166. http://doi.org/10.1016/0167-6377(84)90010-5

Calamai, P., & Moré, J. (1987). Quasi-Newton Updates with Bounds. *SIAM Journal on Numerical Analysis*, *24*(6), 1434–1441. http://doi.org/10.1137/0724092

Chankong, V., & Haimes, Y. Y. (1983). Multiobjective decision making: theory and methodology. *North Holland Series in System Science and Engineering*, (8).

Cho, P. Y. (2011). *Optimal scheduling of fighter aircraft maintenance* (MSc Thesis). Massachusetts Institute of Technology, Cambridge, MA, USA. Retrieved from http://dspace.mit.edu/handle/1721.1/67773

Cohon JL. (1978). *Multiobjective Programming and Planning*. New York: Academic Press.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer.

Ehrgott, M., & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, *22*(4), 425–460.

Florios, K., & Mavrotas, G. (2014). Generation of the exact Pareto set in Multi-Objective Traveling Salesman and Set Covering Problems. *Applied Mathematics and Computation*, *237*, 1–19. http://doi.org/10.1016/j.amc.2014.03.110

Fourer, R., Gay, D. M., & Kernighan, B. W. (2002). *AMPL: a modeling language for mathematical programming*. Duxbury Press, Brooks - Cole Publishing Company.

Gavranis, A., & Kozanidis, G. (2013). An exact solution algorithm for maximizing the fleet availability of an aircraft unit subject to flight and maintenance requirements. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013* (Vol. II, pp. 1036–1041). Hong Kong: Newswood Limited.

Gavranis, A., & Kozanidis, G. (2015). An exact solution algorithm for maximizing the fleet availability of a unit of aircraft subject to flight and maintenance requirements. *European Journal of Operational Research*, *242*(2), 631–643. http://doi.org/10.1016/j.ejor.2014.10.016

Gavranis, A., & Kozanidis, G. (2017). Mixed integer biobjective quadratic programming for maximum-value minimum-variability fleet availability of a unit of mission aircraft. *Computers & Industrial Engineering*, *110*, 13–29. http://doi.org/10.1016/j.cie.2017.05.010

Geoffrion, A. M. (1968). Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, *22*(3), 618–630. http://doi.org/10.1016/0022-247X(68)90201-1

Hahn, R. A., & Newman, A. M. (2008). Scheduling United States Coast Guard helicopter deployment and maintenance at Clearwater Air Station, Florida. *Computers and Operations Research*, *35*(6), 1829–1843.

Helgason, R., Kennington, J., & Lall, H. (1980). A polynomially bounded algorithm for a singly constrained quadratic program. *Mathematical Programming*, *18*(1), 338–343.

Horner, P. (2006, June). Operations Research Management Science Today - June 2006. *OR/MS Today*, *33*(3), 26–28.

IBM ILOG. (2012). CPLEX Optimization Studio: CPLEX User's Manual Version 12 Release 5.

Kozanidis, G. (2009). A Multiobjective Model for Maximizing Fleet Availability under the Presence of Flight and Maintenance Requirements. *Journal of Advanced Transportation*, *43*(2), 155–182.

Kozanidis, G., Gavranis, A., & Kostarelou, E. (2012). Mixed integer least squares optimization for flight and maintenance planning of mission aircraft. *Naval Research Logistics (NRL)*, *59*(3–4), 212–229. http://doi.org/10.1002/nav.21483

Kozanidis, G., Gavranis, A., & Liberopoulos, G. (2008). Heuristics for maximizing fleet availability subject to flight and maintenance requirements. In *10th International Conference on Application of Advanced Technologies in Transportation*. Athens, Greece.

Kozanidis, G., Gavranis, A., & Liberopoulos, G. (2013). Heuristics for flight and maintenance planning of mission aircraft. *Annals of Operations Research*, *221*(1), 211–238. http://doi.org/10.1007/s10479-013-1376-6

Kozanidis, G., Liberopoulos, G., & Pitsilkas, C. (2010). Flight and Maintenance Planning of Military Aircraft for Maximum Fleet Availability. *Military Operations Research*, *15*(1), 53–73.

Kozanidis, G., & Skipis, A. (2006). Flight and maintenance planning of military aircraft for maximum fleet availability: A biobjective model. In *Proceedings of the 18th International Conference on Multiple Criteria Decision Making, Chania, Crete 19-23 June, 2006*. Chania , Greece.

LINGO 13.0. (2011). User's guide, LINDO Systems, Inc., Chicago, IL. Retrieved July 10, 2016, from http://www.lindo.com/

Mattila, V. (2007). Flight time allocation for a fleet of aircraft through reinforcement learning. In *Proceedings of the 2007 Winter Simulation Conference,Washington D.C. , 9-12 December, 2007* (p. 2373). http://doi.org/10.1109/WSC.2007.4419888

Mattila, V., & Virtanen, K. (2005). A Simulation-based Optimization Model to Schedule Periodic Maintenenance of a Fleet of Aircraft. In *Proceedings of the 2005 European Simulation and Modelling Conference, Porto, Portugal, 24 -26 October , 2005* (pp. 479–483). Porto, Portugal.

Mattila, V., & Virtanen, K. (2006). Scheduling Periodic Maintenance of Aircraft through Simulation-Based Optimization. In *Proceedings of the 47th Conference on Simulation and Modelling (SIMS2006), Helsinki, Finland, 28-29 September, 2006* (pp. 38–43). Helsinki, Finland.

Mattila, V., & Virtanen, K. (2014). Maintenance scheduling of a fleet of fighter aircraft through multi-objective simulation-optimization. *SIMULATION*, *90*(9), 1023–1040. http://doi.org/10.1177/0037549714540008

Mattila, V., Virtanen, K., & Raivio, T. (2003). A Simulation Model for Aircraft Maintenance in an Uncertain Operational Environment. In *Proceedings of the 17th European Simulation Multiconference in Nottingham, England, 9 - 11 June, 2003*. Nottingham, England.

Mattila, V., Virtanen, K., & Raivio, T. (2008). Improving Maintenance Decision Making in the Finnish Air Force Through Simulation. *Interfaces*, *38*(3), 187–201. http://doi.org/10.1287/inte.1080.0349

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, *213*(2), 455–465. http://doi.org/10.1016/j.amc.2009.03.037

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*. http://doi.org/10.1016/j.amc.2013.03.002

North Atlantic Treaty Organization (NATO). (2015). AAP-06 NATO Glossary of Terms and Definitions. Retrieved from nso.nato.int/nso/zPublic/ap/aap6/AAP-6.pdf

Pardalos, P. M., & Kovoor, N. (1990). An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, *46*(1), 321–328.

Pippin, B. W. (1998). *Allocating Flight Hours to Army Helicopters* (MSc Thesis). Naval Postgraduate School, Monterey, CA, USA.

Raivio, T., Kuumola, E., Mattila, V., Virtanen, K., & Hämäläinen, R. P. (2001). A simulation model for military aircraft maintenance and availability. In *Proceedings of the 15th European Simulation Multiconference*. Prague, Czech.

Rosenzweig Vojvodić, V., Domitrović, A., & Bubić, M. (2010). Planning of Training Aircraft Flight Hours. In *13th International Conference on Operational Research* (Vol. 1, pp. 170–179). Split, Croatia.

Safaei, N., Banjevic, D., & Jardine, A. K. S. (2011). Workforce-constrained maintenance scheduling for military aircraft fleet: a case study. *Annals of Operations Research*, *1*(186), 295–316. http://doi.org/10.1007/s10479-011-0885-4

Sgaslik, A. (1994). *Planning German Army helicopter maintenance and mission assignment* (MSc Thesis). Naval Postgraduate School, Monterey, CA, USA.

Steiner, A. (2006). A Heuristic Method for Aircraft Maintenance Scheduling under Various Constraints. In *Proceedings of the 6th STRC Swiss Transport Research Conference*. Ascona, Switzerland (Hrsg.).

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons.

U.S. DoA, United States Department of the Army, W. D. (2000). Field Manual No. 3-04.500: Army Aviation Maintenance (Appendix D: Maintenance Management Tools). Retrieved from http://www.globalsecurity.org/military/library/policy/army/fm/3-04-500/index.html

Verhoeff, M., Verhagen, W. J. C., & Curran, R. (2015). Maximizing Operational Readiness in Military Aviation by Optimizing Flight and Maintenance Planning. *Transportation Research Procedia*, *10*, 941–950. http://doi.org/10.1016/j.trpro.2015.09.048

Xidonas, P., Mavrotas, G., & Psarras, J. (2010). Equity portfolio construction and selection using multiobjective mathematical programming. *Journal of Global Optimization*, *47*(2), 185–209. http://doi.org/10.1007/s10898-009-9465-4

Yeung, T. G., Cassady, C. R., & Pohl, E. A. (2007). Mission Assignment and Maintenance Scheduling for Multi-State Systems. *Military Operations Research*, *12*(1), 19–34.

Zhang, W., & Reimann, M. (2014). A simple augmented ∊-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research*, *234*(1), 15–24. http://doi.org/10.1016/j.ejor.2013.09.001