Scheduling and Assignment of Fighter Pilot Training Missions

A Two-Stage Optimization Approach Towards Increasing Personnel Readiness in the Armed Forces

R. Migom

TU Delft

Scheduling and Assignment of Fighter Pilot Training Missions

A Two-Stage Optimization Approach Towards Increasing Personnel Readiness in the Armed Forces

by

Rico Migom

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on September 17, 2021.

Student number:	4992016
Project duration:	June 2020 – Sep 2021
Thesis committee:	Prof.Dr. Dick G. Simons
	Dr.Ir. Alexander H. van Zuijlen
	Dr.Ir. Bram Westerweel
	Dr.Ir. Bruno F. Santos

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

TU Delft

Acknowledgements

Over three years ago, I finally finished an extensive 5-year program to become an officer at the Royal Netherlands Air Force (RNLAF). I was proud to have completed the program and was satisfied that my time as a student had finally come to an end and that I could start working. Or so I thought. After just two months, something in me started itching again: I wanted to go back to university. I had always taken interest in Aerospace Engineering studies taught at the Delft University of Technology (DUT). However, DUT does not provide the opportunity to follow these studies part-time, so combining a study like this with my full-time job would be a challenge. Together with some like-minded friends and colleagues I decided to face the challenge and signed up for the Master's in Aerospace Engineering late 2018. Now, almost three years later, this thesis marks the end of what has been a very interesting and challenging journey.

In general, I would like to express my gratitude to all lecturers, the board of examiners and other personnel of the DUT Faculty of Aerospace Engineering for providing me with the flexibility that I needed to be able to combine this study with my job. Special thanks go out to my supervisors Dr.ir. Bruno Lopes Dos Santos and Dr.ir. Bram Westerweel. Bruno was always available for questions, replying to e-mails even late on sunday nights and his humorous approach helped me to put everything into perspective when this was needed. Bram, my point of contact within the Data Science Cell of the RNLAF, where I conducted my research, acted as a beacon of knowledge and provided me with new insights whenever I got stuck.

Additionally, I would like to thank my friends and family who have supported me through every step of my thesis and the study as a whole. Specifically, I would like to thank my study partner Renze Zijlstra. I would have had a far more challenging time without the numerous study and brainstorm sessions we had together. Lastly, I would like to thank my partner Romy for providing me with the much-needed patience and support.

Rico Migom
Amersfoort, September 2021

Contents

List of Figures

List of Tables

List of Abbreviations

2-FL 2-ship Flight Lead 31, 34, 35, 57, 61

4-FL 4-ship Flight Lead 31, 34, 35, 57, 61

AFIT Air Force institute of Technology 29

B&B Branch-and-Bound 33, 40, 41, 43, 49

BD Benders Decomposition 40, 43

CH Construction Heuristic 29, 30

CPT Captain 34, 35

DSC Data Science Cell xiii

FL Flight Lead 31

FO First officer 34, 35

FOC Fully Operationally Capable 27

FPTSAP Fighter Pilot Training Scheduling and Assignment Problem vii, ix, 29, 37, 43, 49–51, 57, 63–68

GAP Generalized Assignment Problem 32, 37, 38, 43

GRASP Greedy Randomized Adaptive Search Procedure 31, 41–43

IGO "Informatie Gestuurd Optreden" xiii

IP Instructor Pilot 31, 34, 35, 41

LFJ Least Flexible Job 42

LNM Largest Number of Modes 42

LNS Largest Number of Successors 42

LPT Longest Processing Time 42

MIP Mixed Integer Programming 31–34, 37, 38, 40, 43, 45, 49

MiS Minimum Slack 42

MP Master Problem 49, 51

MQT Mission Qualification Training 27

MS Machine Scheduling 42

RCL Restricted Candidate List 41

RNLAF Royal Netherlands Air Force xiii, 27, 31, 32, 35, 46, 51

Introduction

Introduction

The main product of the Armed Forces is operational readiness [28]. In order to achieve personnel readiness, pilots in the Royal Netherlands Air Force (RNLAF) and other Air Forces, have to complete a lot of training events throughout the year. Scheduling these training events is a complex task which is typically done manually, as commercial tools that are used in civil aviation are not applicable in the military sector. The main limiting factors in this process are the availability of pilots, the availability of aircraft and the complexity that comes from flight formations. One of the current developments within the RNLAF is the desired transition to "Informatie Gestuurd Optreden" (IGO), freely translated to Information Driven Operations. In this context, the Air Force is exploring how data science and automation could improve the operating standard within the organisation. Automating complex processes such as the construction of a training schedule and allocation of pilots to this schedule could contribute to this improvement by a twofold increase in efficiency. First, the time spent on constructing the schedules could be reduced. Second, the resulting schedule could make more efficient use of the resources which would lead to higher readiness in the Air Force.

Motivation

In my job, where I work for the RNLAF, I am confronted with the training program of fighter pilots on a daily basis. Consequently, when I started looking for a thesis subject in June 2020, the problem of scheduling training for fighter pilots was one of the first subjects that came to mind. Nowadays, with the Data Science Cell (DSC) the Air Force has a department that focuses on problems like this. After contacting the DSC, it became apparent that this problem was already of their interest. Previous research into this problem has been done by Guljé and Taks ([9], [26]), but both these theses were performed independently of the RNLAF and the models were never implemented. Consequently, the RNLAF and DSC are still looking for a tool that is able to perform the scheduling and assignment of fighter pilot training events.

Problem Statement & Scope

In order to become qualified pilots, to maintain a qualification or to upgrade a qualification, fighter pilots have to complete a certain amount of training events throughout the calendar year. These events are mainly conducted in the form of training missions. Scheduling these training missions is subject to the availability of resources and all training missions have complicating characteristics. The missions should be scheduled and pilots should be assigned to the scheduled missions with the objective to maximize the obtained training value and by doing so maximize the personnel readiness. The scope of this research is to develop a model that solves this problem and produces schedules that result in a higher level of training for the fighter pilots. Ultimately, the model should be accessible to all schedulers and easy to use through a user interface. Furthermore, the model could be integrated with a readily available tool that predicts and regulates aircraft availability in the RNLAF. However, these two aspects are not in the scope of this research. Additionally, external factors that could lead to (partial) cancellation of the missions are not taken into account. Even though the problem originates in the RNLAF fighter jet domain, the goal is that the model should (with a few alterations) also be applicable in other military aviation scheduling problems.

Research Objective & Questions

From an organizational point of view, the desired outcome is that a tool is developed that ultimately leads to higher personnel readiness and that reduces the time spent on constructing schedules. From this organizational objective and the problem statement, the research objective is formulated. The research objective for this thesis is to develop a model that is able to schedule initial, recurrent and transition training missions and assign pilots to these missions for training programs that have to be completed within one year. Implementing this model at military fighter squadrons should result in more efficient schedules that lead to an improvement in the readiness of personnel, while also reducing the time spent on constructing these schedules. To the best of our knowledge after having conducted an extensive literature review, no such model currently exists. To meet the research objective and work towards this objective in a structured manner, the following research question and sub-questions have been formulated.

> How can we develop a model that is able to schedule fighter pilot initial, recurrent and transition training missions and assign pilots to these missions in such a way that the readiness of the pilots is maximized?
>
> - What are the current challenges in scheduling training flights within a typical fighter squadron?
> - What are the requirements of the initial, recurrent and transition training programs?
> - What resources are available and how is the planning environment constrained?
> - What model form is suitable to find a solution to this problem within the specified requirements?
> - What methodology is suitable to solve the model and balance runtime and solution quality?

Thesis Outline

The structure of this thesis report is as follows. In Part I, the scientific paper is included. This paper is the core of the thesis and can be regarded as a stand-alone document. In Part II, the Literature Study which preceded the thesis is included. This Literature Review was already presented and graded during an earlier course, but provides significant background information on the subject. Part III provides supporting material, which can serve as a more in-depth explanation of certain aspects of the scientific paper. Lastly, Part IV contains the Appendices.

I

Scientific Paper

Scheduling and Assignment of Fighter Pilot Training Missions

Rico Migom,*

Delft University of Technology, Delft, The Netherlands

Abstract

Military fighter pilots spend most of their working hours performing training missions, simultaneously carried out by multiple pilots with different qualifications. Scheduling these missions is typically done manually, which is a time-consuming task resulting in far from optimal schedules. This study proposes a model that schedules three types of training missions and assigns pilots to these missions for a full-year training program. The proposed methodology consists of a two-stage Mixed Integer Program. Both stages are solved with a Brand and Bound algorithm. Additionally, the second stage implements a Rolling Horizon approach. A case study considering two Royal Netherlands Air Force fighter squadrons is used to test the model. The results show an increase of more than 15% for the average amount of training completed by all pilots, compared to the current scheduling process. The solutions are obtained in a few minutes, demonstrating the suitability to use the model in practice and reduce the time spent on constructing schedules.

Index Terms: Military Aviation, Scheduling & Assignment, Rolling Horizon, Mixed Integer Programming, Royal Netherlands Air Force

1 Introduction

According to a Dutch policy research proceeding *Zicht op Gereedheid*, the most important product of the armed forces is operational readiness, the ability to (inter)nationally deploy trained units and weapon systems [24]. One of the key aspects of operational readiness is the readiness of personnel. To achieve maximum readiness of personnel, the armed forces strive to offer all their personnel as much training as possible. This is especially true for fighter pilots, who undergo a lengthy and intensive initial training program. But even after pilots have completed the initial training, much of their yearly flying program consists of performing training missions with the aim of maintaining or increasing their level of proficiency. To do so, different training programs exist that have to be completed in the course of a calendar year. These training programs describe a set of specific training missions that have to be flown by a pilot to acquire or extend a pilot's license or upgrade the qualification of the pilot. While scheduling these training missions, the available resources as well as additional constraints have to be taken into account. These constraints are for example the result of flight formations. For the remainder of this paper we call the problem of scheduling training missions and assigning pilots to these missions given the constraints and available resources the Fighter Pilot Training Scheduling and Assignment Problem (FPTSAP). The organizational objective for the FPTSAP is then to maximize the readiness of personnel.

In practice, the training programs that have to be followed by the pilots, can be categorized into three types of training: initial training, recurrent training and transition training. Initial training is aimed at Student Pilots (SPs) that have just arrived at the squadron. They need to complete a training program that is strictly sequentially ordered and the SPs have to be supervised by an instructor during each mission. Recurrent training on the other hand is focused on readily qualified pilots. Those pilots have to conduct a certain amount of training missions in order to maintain their qualification. Lastly, transition training is meant to have qualified pilots upgrade their qualification to a higher level. As a result, transition training is in this paper also called upgrade training. Like initial training, transition training has to follow a sequentially ordered syllabus. All three types of training have a different impact on the personnel readiness, so managerial decisions could be made to prioritize one training type over another.

In the FPTSAP, a mission is defined as a training event that is conducted by a predetermined number of pilots. Missions that have similar goals and characteristics are grouped together in several mission categories. If a mission has to be flown by four, two or one pilot(s), the formation is said to be respectively a 4-ship, a 2-ship or a singleton. On top of that, some missions require opposing forces. Those opposing forces are scheduled simultaneously as red air missions (support), while the actual training missions are defined as blue air. Then, every pilot has a qualification, stating which roles the pilot can assume in every formation. Qualifications typically used in North Atlantic Treaty Organization (NATO) Air Forces and European Participating Air

*MSc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

Forces (EPAFs) are, from highest to lowest qualification, Instructor Pilot (IP), 4-ship Flight Lead (4-FL), 2-ship Flight Lead (2-FL), Wingman (WM) and SP. Pilots with higher qualifications automatically also hold all lower qualifications, except for SP. Depending on the qualification, level of experience and strategic decisions, every pilot is assigned one or more training syllabi. A syllabus can then be defined as a specific combination of missions that a pilot has to complete during the scheduling horizon. The scheduling horizon is then split up into multiple weeks and working days. A working day can be divided into multiple periods during which missions can be scheduled. Such a period is called a go. The process of a single pilot (in a fighter jet) that is conducting a mission is defined as a sortie. Additional complexity comes from the fact that pilots conducting a sortie in the same mission can do so as part of different syllabi. An overview of how these aspects come together in a schedule is given in Table 1. To get an idea of the complexity of scheduling for just one go, we take a look at this example. Two missions are scheduled in go 'AM 12', mission 19 and mission 2. Mission 19 is part of mission category A and is carried out by six aircraft and pilots simultaneously, divided over a 4-ship formation for blue air and a 2-ship formation for red air. In the 4-ship at least one pilot with qualification 4-FL or higher and another pilot with qualification 2-FL or higher should be present. The other two pilots normally need to be of qualification WM or higher. If an SP is assigned to the mission as part of his initial training syllabus, like pilot 14 in this case, the SP has to be supervised by an IP, pilot 1. At the same time, pilot 1 can perform this mission as part of his recurrent training. However, the pilots performing this mission as red air, can not count the mission towards any of their training programs.

Table 1: Partial example of a training schedule.

Week	**Go**	**Mission**	**Mission Category**	**Pilot**	**Role**	**Qualifi-cation**	**Syllabus**
3	AM 12	19	A	1	Blue	IP	Recurrent
		19	A	14	Blue	SP	Initial
		19	A	5	Blue	2-FL	Recurrent
		19	A	10	Blue	WM	Recurrent
		19	A	2	Red	2-FL	Support
		19	A	8	Red	WM	Support
		2	B	7	Blue	2-FL	Recurrent
		2	B	11	Blue	WM	Recurrent
	PM 12	...	...	...	...	...	

Considering these aspects, the FPTSAP is a heavily constrained and complex scheduling problem. Nevertheless, scheduling is typically done manually as a secondary task by (ex-)fighter pilots with little scheduling experience. Additionally, planning is regularly done with few (if any) decision support tools available to the schedulers. As a consequence, scheduling is a time consuming task. To simplify the process and prevent infeasible schedules, schedulers are often forced to shorten their scheduling horizon. The combination of manual scheduling and short scheduling horizons leads to schedules that are far from optimal. Several studies have already attempted to improve the process of scheduling fighter pilot training missions and assigning pilots to these missions, but all fall short in one or more ways. First, the papers by Aslan and Nguyen ([1], [14]) only focus on the initial training program. Second, the proposed solutions of Newlon, Yavuz and Erdemir ([13], [26], [4]) either solve their scheduling problems with outdated methodologies or have a scheduling horizon of only one day or week. The master thesis of Taks proposes a model that solves the FPTSAP for a full year, but it lacks detail and constraints with respect to formations, initial training, transition training and training completion per pilot [23]. Additionally, due to the absence of a real case study it is hard to verify the performance of the model. The methodology proposed in the master thesis by Guljé leads to runtimes of up to seven hours and low readiness of personnel, while only scheduling recurrent training [8].

The research objective for this paper is to develop a model that is able to schedule initial, recurrent and transition training missions and assign pilots to these missions for training programs that have to be completed within one year. Implementing this model at military fighter squadrons should result in more efficient schedules that lead to an improvement in readiness of personnel, while also reducing the time spent on constructing these schedules. The model should also be able to react to changes in resource availability or training demand throughout the scheduling horizon. To reach this objective, this paper proposes a two-stage Mixed Integer Program (MIP) model. The first stage constructs a preliminary schedule, where each week of the scheduling horizon is assigned a mission category. This stage takes into account the demand for certain missions, the number of goes available per week and the number of aircraft available per week. Subsequently, the second stage schedules specific missions and assigns pilots per week using a Rolling Horizon (RH) approach. While doing so, the model has to take into account the availability of pilots with specific qualifications, availability of aircraft, formation requirements, demand for certain missions and limitations of pilots in initial or transition training. Both models are solved with the Branch-and-Bound (B&B) algorithm of a commercial solver. To test the model, it is subjected to a case study which originates in the Royal Netherlands Air Force (RNLAF), an Air Force that resides among the best trained and most advanced Air Forces in the world.

The structure of this paper is as follows. Literature related to the problem is discussed in Section 2. Section 3 provides an explanation of the proposed methodology. The case which is used to test and demonstrate the model is described in Section 4. Subsequently, Section 5 presents the results from the case study and a sensitivity analysis. Finally, the conclusion and resulting recommendations are presented in Section 6.

2 Literature Review

In this Section, we explore existing literature that focuses on the scheduling and assignment of aircrew training. By doing so, we aim to identify the current level of understanding in this field of research, how several studies interrelate and where there is a possible gap in the research. We distinguish between civil and military aircrew training scheduling and assignment problems, as these two types are very different from one another.

In the military aviation sector, the literature that already exists can further be divided into two categories. Firstly, there is the literature that focuses solely on initial training. Both the works of Aslan and Nguyen ([1], [14]) research such a problem. They attempt to build a decision support tool to support planners in scheduling and assigning initial training at a fighter training squadron. These researches are relevant for this paper, because the problem environment concerning initial training is the same as the problem described in the introduction. Their methodologies however are outdated and rely on manual input by the scheduler throughout the scheduling process. Secondly, there are several papers and theses which discuss recurrent training. Scheduling recurrent training is more complex for two main reasons. First, there is often no fixed sequential syllabus. This leads to a larger solution space when compared to problems that only consider initial training. Second, the training missions in recurrent training often have to be flown by large formations and can require additional supporting aircraft. Research on such scheduling problems was carried out by Newlon, Yavuz, Erdemir, Taks and Guljé ([13], [26], [4], [23], [8]). Newlon's research is a recurrent training scheduling and assignment problem, with a scheduling horizon of one week. The main goal of the operations within the squadron is focused on maintaining pilot's licenses. Similar to Nguyen and Aslan earlier, Newlon's goal is to find a feasible schedule in reduced time. The model developed by Newlon solves in a matter of minutes, but assumes unlimited resources. Furthermore, the three pilot qualifications he uses could be expanded further [13]. Yavuz' research takes a readily available weekly schedule and only assigns pilots to recurrent training missions [26]. The main objective in this study is that all pilots should maintain their licenses and perform some yearly mandatory flights. The main benefits of the developed method over fully manual scheduling are that the total amount of training events is balanced better between the pilots and that feasible assignments are found faster. The downside to this work is that it is an assignment problem rather than an integrated scheduling and assignment solution [26]. Erdemir studies a recurrent training scheduling and assignment problem, with a scheduling horizon of one day [4]. The main objective is that all pilots should maintain their pilot's license, while maximizing sorties. For small instances, the model is able to find optimal solutions within seconds. For realistically sized data samples using twenty pilots however, the runtime increases to more than one hour. As such, Erdermir suggests to further investigate heuristics to reduce runtime. If reduced runtime can be realized, he also suggests to increase the scheduling horizon [4]. Taks increases the scheduling horizon to a full year. He researches a case where both initial and recurrent training have to be scheduled and assigned [23]. Taks aims to provide squadron schedulers with a baseline reference schedule to ease the schedulers' decision making process. He succeeds to find a yearly schedule within several minutes, but a lot of simplifications have been made to reduce the complexity of the problem. On top of that, when the required number of sorties increases the model is no longer able to find feasible solutions [23]. Guljé studies almost the same problem as Taks, but adds more complexity to the problem and runs the yearly scheduling and assignment process as a simulation [8]. The objective of his model is to have as many pilots as possible achieve full completion of the yearly training program. Computation times are in excess of multiple hours and completion of the training program is, as stated by Guljé himself, under par. He believes further research could be dedicated towards simplifying the model while at the same time looking into aircraft availability [8].

Aside from these studies on scheduling and assignment of training in military aviation, there also exists relevant research on the subject of pilot training in the civil aviation sector. Examples are the work done by Holm, Kempen, Kozanidis, Qi et al. and Yu et al. ([10], [25], [12], [18], [27]). From these papers we conclude that these civil problems as a whole are very different in terms of scheduling environment, resources and constraints. The differences are mainly a consequence of the lack of flight formations in civil aviation. Nevertheless, some of the research provides useful insights on aspects of the FPTSAP such as pilot qualifications.

In both the civil and military aviation literature, the fundamental methodology to formulate pilot training scheduling problems is MIP. In MIP an objective function is defined, which is in turn maximized while respecting a set of constraints and predefined parameters. Applying MIP to real size, real world problems however, often results in large problems with rapidly increasing computation times. In fact, crew scheduling is known as an

NP-hard problem [3]. As such, crew scheduling problems and their MIP models are often reformulated in order to improve computational efficiency. One way to formulate a MIP is the Generalized Assignment Problem (GAP), introduced by Ross [20]. GAP in its most basic form is a straightforward scheduling problem, in which a set of tasks has to be performed by a set of agents. Both Yavuz and Guljé ([26], [8]) link fighter pilot scheduling to the GAP. In the civil domain, all of Holm, Kozanidis, Van Kempen, Qi et al. and Yu et al. use a form of the GAP ([10], [12], [25], [18], [27]). A second option is to divide the problem into multiple subproblems, but this simple approach also has a downside. The creation of subproblems might have as a consequence that an optimal solution to the overarching problem can no longer be found. One way to split a main problem into subproblems is by defining the subproblems in such a way that they can be solved simultaneously. A simpler method is to split the main problem into several subproblems which can then be solved sequentially. This methodology has been used by Aslan, Erdemir, Taks and Yu et al. ([1], [4], [23], [27]) The RH approach, explained by Silvente et al. [22], could be seen as a specific case of solving a model in several sub-problems. Using a RH approach, the scheduling horizon is split into discrete time periods. At each iteration, the problem is solved for the current time period but it can take into account information from past time periods or forecast data from upcoming time periods. RH approaches have been used by Qi et al., Yu et al. and Guljé ([18], [27], [8]). Another way to simplify a problem is to apply set partitioning to the problem as defined by Garfinkel and Nemhouser [7]. Set partitioning can reduce the problem size significantly and can often be applied by using common sense. For example, in an aircrew training scheduling and assignment problem one could apply set partitioning to the pilots by dividing them into different sets in accordance with their qualifications as has been done in almost all the research discussed.

A widely applied methodology to solve MIP problems is B&B, accurately explained by Clausen [3]. Clausen states that *"B&B is by far the most widely used tool for solving large scale NP-hard combinatorial optimization problems."*[3] A B&B algorithm is able to search the whole solution space of a problem. Considering the previously discussed literature, it becomes apparent that B&B is indeed a popular tool; it is applied in the research of Holm, van Kempen, Qi et al. and Yu et al. ([10], [25], [18], [27]) Even though commercial solvers use highly complex and efficient B&B algorithms, solving large-scale problems to an exact and optimal solution often still results in excessive runtimes. To overcome this problem, heuristics have been created. A definition of heuristics can be found in the work of Salcedo-Sanz [21], but in short heuristics can be described as smart tricks that attempt to decrease the computation time in which an acceptable solution can be found. For example, a model developed by Kim and Kim needs three hours to solve using exact optimization methods, but solves within seconds when applying their heuristic [11]. Examples of heuristic methods applied in scheduling and assignment problems of our interest are Greedy Randomized Adaptive Search Procedure (GRASP) and dispatching rules ([19], [17]). GRASP has been used in the FPTSAP problem by Yavuz and by Erdemir ([26], [4]). Dispatching rules have been used in fighter pilot scheduling by both Aslan, Nguyen and Guljé ([1], [15], [8]).

From the literature overview, it can be concluded that there are no publications that present model solutions capable of both scheduling and assignment of recurrent, initial and transition training in a military environment for a scheduling horizon of up to one year. The apparent gap in literature motivates this particular research project. This research proposes a model solution that is able to close this research gap. The contributions of the proposed solution methodology are the following:

- The two-stage solution methodology makes the model useful for scheduling horizons of one week up to one year, so the model can be used in multiple phases of the planning process.
- The proposed model takes into account scheduling and assignment of all three of initial, recurrent and transition training.
- The proposed model can solve for a scheduling horizon of one year within minutes.
- An option is included in the model that allows pilots to train for and go on deployments.
- The model is developed to be applicable to fighter operations in all NATO Air Forces and EPAFs. On top of that, with a few minor alterations, the model should also be applicable to other weapon systems.

3 Model Formulation

This Section provides an overview and explanation of the methodologies that are used to formulate and solve the problem. First, some additional info is given on the model requirements and assumptions that are made. Second, in Subsection 3.2, the high-level solution methodology is explained. An explanation of the required input data is given in Subsection 3.3. Lastly, Subsections 3.4 and 3.5 provide a description of the two models that are used to solve the FPTSAP.

3.1 Model Requirements & Assumptions

The proposed model should be able to schedule all three types of training. Strategic decisions could be made that would result in a higher priority for either one of the three training types. For example, if there are few 4-FLs and many 2-FLs the decision could be made to prioritize transition training to have some 2-FLs upgrade to 4-FLs quickly. To reflect these types of decisions, the model should be able to assign priority to either of the three training types.

In addition to the three standard types of training, the model should also be able to handle deployments. A *deployment* is defined by the United States Department of Defense as *"The movement of forces into and out of an operational area."* [16]. The impact of a deployment in this model is that certain pilots can be appointed to go on a deployment. These pilots have to complete a certain training program before the deployment starts and are unavailable for other activities during the length of the deployment.

It is impossible to capture every detail of the real-life problem in a model. However, as was stated by Box, *"All models are wrong, but some are useful."*[2] Therefore, in order to make it possible to clearly formulate a mathematical model some assumptions have to be made. These assumptions are:

1. Every week, only missions belonging to one specific mission category can be scheduled.
2. Only one mission can be completed per sortie.
3. All missions and accompanying sorties that are scheduled will be successfully executed.
4. Formation sizes are either singleton, 2-ship or 4-ship.
5. Only one type of aircraft is considered when constructing the schedule.
6. The number of aircraft that are available for scheduling at any moment is known for the whole scheduling horizon at the beginning of the scheduling horizon.
7. The number of pilots and the availability of all pilots is known for the whole scheduling horizon at the beginning of the scheduling horizon.

These assumptions closely resemble RNLAF practices, except for assumptions 1 and 7. Assumption 1 states that one mission category can be scheduled per week. In practice, missions within the same category are often scheduled in the same week without this being a strict rule. The continuity that would come from such a rule however could benefit pilots, ground crew, schedulers and the organization as a whole as it would make the whole training process easier. Assumption 7 states that the availability of all pilots is known for the full scheduling horizon. It is impossible to predict the exact availability, but personnel is advised to request their leave as early as possible. Although it is hard to exactly estimate the resources availability, it is fair to assume that the availability is known since the personnel leaves are commonly known a few months in advance. Even if significant changes occur in personnel availability, it becomes clear in Section 5 that the model runtime is low enough to be able to quickly build a new schedule.

3.2 High-level Methodology

The goal of the FPTSAP is to efficiently schedule training missions over the goes and to assign sorties to pilots in such a way that every pilot can complete as much of the assigned training as possible. Eventually, this leads to higher operational readiness. To achieve this goal, the decision is made to split the problem into two stages: a Master Problem (MPr) and multiple Subproblems (SPrs). The MPr looks at the full scheduling horizon. For each week, it can either schedule one mission category or no mission category at all. One would want to minimize the number of weeks that are used in the scheduling horizon because other activities, like large exercises, also make use of these weeks. The goal of the MPr is to make just enough sorties available for every mission category, so that every pilot can perform the required number of missions in that mission category. This schedule is then forwarded to the first SPr. Every SPr has a scheduling horizon of one week and as such can only plan missions in the scheduled mission category for that week. The SPrs schedule these missions and assign pilots to these missions. The SPrs have the objective to maximize the amount of training, or to increase the readiness, that is obtained every week. There are two reasons to approach the FPTSAP like this. The first reason is computational efficiency. During an early stage of the research, we tried to solve the problem as a whole. Due to the size of the problem and the NP-hard nature of the problem, we were not able to find any solution within a day and therefore decided that a more efficient methodology should be developed. The second reason comes from an organizational point of view. The MPr can provide a preliminary schedule for the full scheduling horizon within seconds and without going into detail. This schedule can serve as a starting point or be used to make high-level decisions within parts of the organization that need to look further into the future.

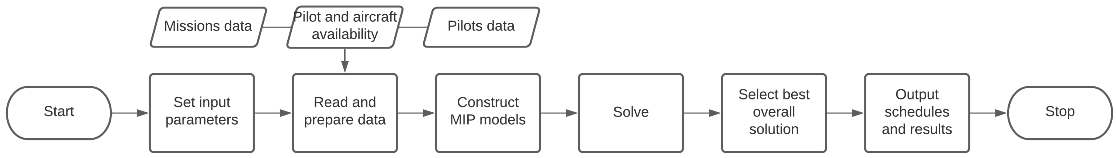


Figure 1: Flow chart of the high level solution process of the FPTSAP.

The SPrs then go into more detail and can be used for the lower-level decision making process. This two-stage methodology thus makes the model useful throughout multiple levels of an organization.

Figure 1 visualizes the flow of the model as a whole. The first step is to set the user-defined input parameters. The most important parameters are the length of the scheduling horizon, the number of goes in a week, the desired level of training, the availability of pilots and the availability of aircraft. All parameters can be found in Tables 5 and 8. After this has been done, the data that dictates which pilots are in the problem and how many times they should perform every mission is read by the model and structured to run the model. The exact composition of this data is described in Section 3.3. The next step is to construct the MIP models that represent the MPr and SPrs. Subsequently, the problem is solved according to the process visualized in Figure 2. After terminating the solution process, the best solution to the overall FPTSAP problem is selected. For this solution, the results are processed and a schedule is generated. An example of a scheduled mission and assigned pilot is AM1: 4, 4-F4-2. This means that at the AM go of day 1, pilot 4-F4-2 is assigned to mission number 4. The pilot discriminator 4-F4-2 denotes the fourth pilot overall and the second pilot with qualification 4-FL. The output schedules of the model, an example of which can be found in Appendix A, are anonymized for confidentiality reasons.

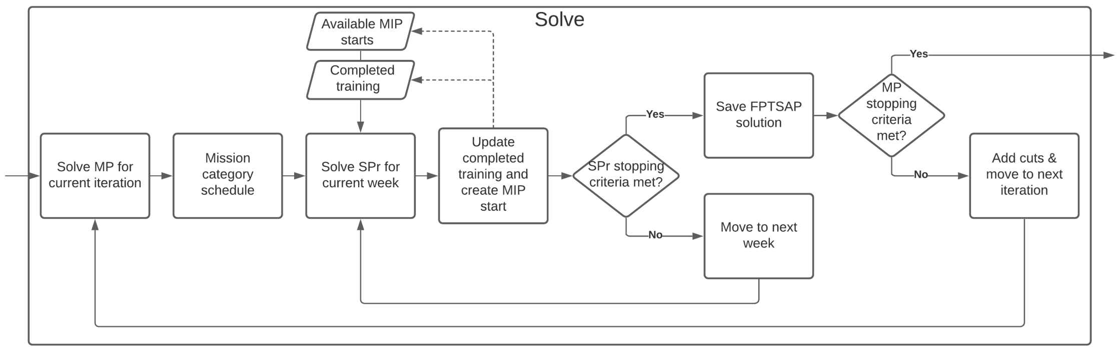


Figure 2: Flow chart of the Solve process from Figure 1.

The solution process given in Figure 2 starts by solving the MPr. The MPr is a MIP that focuses on the full scheduling horizon. It splits this scheduling horizon into separate weeks and schedules one mission category for every week. Thus, if a mission category is assigned to a week, only missions that belong to that specific mission category can be scheduled in that week. Ultimately, the MPr aims to fulfill the training demand in the minimum amount of weeks, by either assigning a specific mission category or not assigning any mission category to each week. The specifics of the MPr are discussed further in Subsection 3.4. The MPr is solved by Gurobi, a commercial solver that uses a combination of B&B and heuristic methods to solve optimization problems [9]. The resulting output is the mission category planning per week, which is forwarded to the SPrs. Every SPr is also a MIP, with a scheduling horizon of one week. As such, a single full solution process consists of as many SPrs as there are weeks in the MPr scheduling horizon. These separate SPrs are solved with a RH approach. Every SPr has the goal to schedule missions and assign sorties in such a way that the highest training value can be attained every week in accordance with the mission category that was scheduled by the MPr and the training already completed in previous SPrs. The specifics of the SPr are discussed further in Subsection 3.5. Each SPr is solved by Gurobi. At the end of each week, the results of that week are saved and carried over to the beginning of the next SPr. The next step is to check if any SPr stopping criteria is met. For this paper the only stopping criterion is reaching the end of the scheduling horizon, but other criteria could be implemented. If none of the implemented stopping criteria is met, the model continues to the next week, where a new SPr is solved. When one of the criteria is met, the solution to the entire FPTSAP, so the solution for the full scheduling horizon is saved. Continuing the process, the model evaluates if any of the stopping criteria of the MPr has been met. Stopping criteria for the MPr are either reaching a predetermined amount of iterations or achieving a predetermined level of training. The iterative process is introduced because of the nature of the MPr. The MPr solves by looking for the minimum amount of weeks in which the desired training value can be met. However, different solutions in terms of mission category planning per week could lead to the same minimum amount of weeks required, so the first solution to the MPr is not necessarily the best one. Therefore, if no stopping criteria is met, a cut is implemented to reduce the feasible solution space. In the current formulation, there is only one type of cut present. This cut forbids solutions that offer just as many available goes for every mission category as any previous solution did. After adding the cuts, the process restarts by solving the MPr. If no MPr stopping criteria was met in the step before, the model exits the solve block.

This full process visualized in both Figures 1 and 2 is also described with pseudo-code in Appendix B.

3.3 Input Data

The algorithm that is detailed in Subsections 3.2, 3.4 and 3.5 requires specifically structured data in order to be run. This subsection provides an explanation of how this data should be structured.

Pilot Data The pilot data contains information on the number of pilots in the scenario, their qualification, their status and the training syllabi that have been assigned to them. This data is anonymized, but resembles the actual situation at a RNLAF fighter squadron. A partial example of the data is given in Table 2. The column *P ID* is an indexed identifier for each pilot, that starts at 1. *Qualification* states the highest qualification the pilot holds. Known qualifications to the model are IP, 4-FL, 2-FL, WM and SP. Qualified Pilots (QPs) are all pilots except SP. The next column, *Status*, states if the pilot is either experienced (exp) or inexperienced (inexp). This is based on how long the pilot has already been a QP for and can have consequences for the amount of training sorties required. Lastly, the column *Syllabi* notes for each pilot which Syllabi are assigned. Normally, all QPs are assigned recurrent training (RT) and all SPs are assigned initial training (IL). On top of that, a WM can be assigned transition training to become a 2-FL (U2) and a 2-FL can be assigned transition training to become a 4-FL (U4). Additionally, all QPs can be assigned a deployment work-up syllabus (DY) to prepare them for a deployment.

Table 2: Example of pilot data.

P ID	**Qualification**	**Status**	**Syllabi**
		...	
14	4-FL	exp	RT
		...	
18	2-FL	exp	RT, U4
19	2-FL	inexp	RT
		...	
23	SP	inexp	IL

Mission Data The mission data includes all available missions and their characteristics. This information is obtained from the *RNLAF F-16 training program* and the *RNLAF Mission Qualification Training* ([5], [6]), but has been made unrecognizable for classification reasons. An example of some missions and their characteristics is given in Table 3. All missions are numbered with a unique index, *M ID*. The syllabi in which they occur are given in column *S*. Then the number of pilots required as blue air is given in column *Blue size*. Some missions require red missions to be activated at the same time. *Red mission* indicates the *M ID* of the red mission that should be scheduled. The next column, *#R1* indicates how many times the mission has to be executed in the recurrent training syllabus for experienced pilots. Similar columns exist for recurrent training for inexperienced pilots, initial training, transition training and deployment work-up, but these are left out for better readability. Then column *Prec* states the indices of the required precedents for the mission, if there are any. Missions in the upgrade syllabi from WM to 2-FL (U2) and from 2-FL to 4-FL (U4) always are similar to a mission in the recurrent syllabus. For the functionality of the model however, these upgrade missions need a unique mission index. The recurrent training mission they resemble is given in the column *Twin*. Column *MC* indicates which mission category the mission belongs to and *MC Alt* states an alternate mission category in which the mission might also be scheduled.

3.4 Master Problem

The idea behind the MPr is that it decides for every week in the scheduling horizon which mission category to assign, if any. The MPr bases these decisions on the demand for missions in each mission category, given

Table 3: Example of mission data.

M ID	S	Blue size	Red mission	#R1	...	Prec	Twin	MC	MC Alt
					⋮				
5	IL, RT	2	35	1	...	4	/	A1	/
					⋮				
12	IL, RT	4	36	2	...	9	/	A4	/
13	IL	2	35	0	...	6	/	A2	/
					⋮				
51	U2	4	36	0	...	42, 44, 45	12	A4	G3

the pilots and their assigned syllabi. By doing so, the MPr aims to provide the SPrs with enough scheduling opportunities to meet the required readiness in as few weeks as possible. The Sets, Parameters and Decision Variables used in the MPr are described in Tables 4, 5 and 6 respectively.

Table 4: Sets used in the MPr.

Set	Description
$\mathbf{MC}$	set of mission categories
$\mathbf{W}$	set of weeks
$\mathbf{DY}$	set of deployments
$\mathbf{PRC}_{mc}$	Set of precedents of missions category mc

Table 5: Parameters used in the MPr.

Parameter	Description
bw_{dy}	week in which deployment dy starts
$demg_m$	total demand for mission m in terms of goes
$dems_m$	total demand for mission m in terms of sorties
gpw	number of goes in a week
na_w	number of aircraft available in week w
rdy_m	number of required executions of mission m to be ready for deployment
τ	training value threshold

$$\min \quad \mathcal{J} = \sum_{w\in\mathbf{W}} \sum_{mc\in\mathbf{MC}} vmcs_{w,mc} \tag{1a}$$

s.t.

$$\sum_{mc\in\mathbf{MC}} vmcs_{w,mc} \leq 1, \qquad \forall w \in \mathbf{W}, \tag{1b}$$

$$\sum_{w\in\mathbf{W}} vmcs_{w,mc} \cdot na_w \cdot gpw \geq \tau \cdot \sum_{m\in\mathbf{M}_{mc}} dems_m, \qquad \forall mc \in \mathbf{MC}, \tag{1c}$$

$$\sum_{w\in\mathbf{W}} vmcs_{w,mc} \cdot gpw \geq \tau \cdot \sum_{m\in\mathbf{M}_{MC}} demg_m, \qquad \forall mc \in \mathbf{MC}, \tag{1d}$$

$$\sum_{w=1}^{bw_{dy}} vmcs_{w,mc} \cdot na_w \cdot gpw \geq \sum_{m\in\mathbf{MC}} rdy_m, \quad \text{if } m \in [\mathbf{M}_{dy} \cap \mathbf{M}_{mc}], \quad \forall mc \in \mathbf{MC}, dy \in \mathbf{DY}, \tag{1e}$$

Table 6: Decision variables used in the MPr.

Variable	Domain	Description
$vmcs_{w,mc}$	$\{0, 1\}$	1 if mission category mc is scheduled for week w

$$\sum_{w \notin \mathbf{W}_{dy}} vmcs_{w,mc} \geq 1, \qquad \forall mc \in \mathbf{MC}, dy \in \mathbf{DY}, \tag{1f}$$

$$vmcs_{w,mc} \leq \sum_{v=1}^{w-1} vmcs_{v,mcp}, \qquad \forall mc \in \mathbf{MC}, mcp \in \mathbf{PRC}_{mc}, w \in \mathbf{W}. \tag{1g}$$

Objective Function The MPr objective function is to minimize the amount of weeks in the schedule that get assigned a mission category by minimizing the sum of the binary decision variable $vmcs_{w,mc}$ for all weeks and mission categories, as can be seen in Equation 1a.

Constraints To ensure that enough weeks are made active in order for the SPrs to schedule sufficient training, some constraints have to be implemented. Constraints 1b state that at most one mission category can be active per week, as was noted in assumption 7 in Subsection 3.1. Constraints 1c and 1d enforce that enough weeks are activated in order for the SPrs to be able to fulfill a certain ratio of the training demand and thus to achieve a certain level of readiness, while taking into account the available aircraft and goes per week. $\mathbf{M}_{mc}$ is a subset of $\mathbf{M}$ that contains all missions in mission category mc. Constraints 1e and 1f apply to deployments. The first states that mission categories that contain missions that have to be completed before a pilot is eligible to go on deployment should be scheduled often enough before the start of the deployment. Set $\mathbf{M}_{dy}$ is a subset of $\mathbf{M}$ that contains the missions that are part of the deployment work-up. The second states that every pilot that goes on a deployment must get the chance to train every mission category at least for one week, outside of his deployment period, where $\mathbf{W}_{dy}$ denotes the sets of week for each deployment period. Constraints 1g make sure that mission categories that have missions with precedents in other categories are scheduled in a logical order. These constraints prevents weeks where no initial or transition training can be scheduled. The most important results and output of the MPr are then the total number of active weeks and the mission category planning per week, which is forwarded to the SPrs.

3.5 Subproblem

The mathematical formulation for the SPrs is given in this Subsection. For clarification, it is noted that sets presented in Table 7 can have subsets that originate from other sets. For example, if there exists a pilot qualification SP, there exists a set of pilots that own qualification SP. This set of pilots, that have qualification SP, which is a subset of $\mathbf{P}$ is denoted as $\mathbf{P}_{SP}$. For readability purposes, a set with multiple subscripts separated by a comma is used to denote the intersection of two subsets, so $\mathbf{M}'_{4,U2}$ indicates the set of missions of size 4 in the 2-FL transition training syllabus. Furthermore, every SPr has a scheduling horizon of only one week and the solution from the MPr states which mission category is active in this week. The apostrophe that is displayed next to sets denotes that we are only looking at that part of the set that is relevant for the current SPr. For instance, $\mathbf{G}'$ means all the goes in the current week and $\mathbf{M}'$ are all missions in the mission category that is scheduled for this week. Parameters used by the SPr are explained in Table 8. The most important parameters are the number of sorties per mission that is required and already completed by each pilot.

Decision Variables The decision variables for the SPr are given in Table 9. It is useful to note the difference between $vs_{m,g,p}$ and $vsu_{m,g,p}$. All pilots are either in initial training or in recurrent training, so the model can differentiate between those types of training by looking at different sets of pilots. Some pilots that are in recurrent training however, are also assigned transition training. Those pilots can be assigned training sorties that relate to either of both syllabi. However, the model needs to be able to differentiate whether the pilot is performing the mission either as part of the recurrent training program or as part of the transition training program. If pilot p performs mission m at go g as part of the recurrent syllabus, this sortie will count towards variable $vs_{m,g,p}$. If the pilot performed this mission as part of the transition syllabus, the sortie will count towards variable $vsu_{m,g,p}$. Every go, $vm_{m,g}$ determines how many times every mission is scheduled. The ratio of completion for every recurrent training mission by each pilot is tracked by $vtmp_{m,p}$. This variable serves as an auxiliary variable to be able to compute vtp_p. The two training completion ratios that are passed to the objective function are vtp_p and $vtpu_p$. The training completion values could be fed directly to the objective function without the introduction of vtp_p and $vtpu_p$. However, for readability reasons and ease of programming,

Table 7: Sets used in the SPr.

Set	Description
G	set of goes
P	set of pilots
M	set of missions
DY	set of deployments
Q	set of pilot qualifications
SY	set of training syllabi
BSI	set of blue air mission sizes
$\mathbf{PRC}_m$	Set of precedents of mission m

Table 8: Parameters used in the SPr.

Parameter	Description
bw_{dy}	week in which deployment dy starts
$c_{m,p}$	number of executions of mission m by pilot p
m_{red}	red air missions that has to be planned simultaneous to mission m
na_w	number of aircraft available in week w
$av_{p,g}$	availability of pilots; 1 if pilot p is available at go g
$r_{m,p}$	number of executions required by pilot p for mission m
$rdy_{m,p}$	number of executions required by pilot p for mission m to be ready for deployment
w_i	weight factor for initial training missions
w_r	weight factor for recurrent training missions
w_u	weight factor for upgrade training missions

the choice is made to introduce these decision variables. A similar relation as to $vs_{m,g,p}$ and $vsu_{m,g,p}$ exists between the two decision variables vtp_p and $vtpu_p$.

Table 9: Decision variables used in the SP.

Variable	Domain	Description
$vs_{m,g,p}$	$\{0,1\}$	1 if pilot p is assigned to mission m at go g
$vsu_{m,g,p}$	$\{0,1\}$	1 if pilot p is assigned to mission m, which is part of an upgrade training, at go g
$vm_{m,g}$	$\{0,1\}$	number of times mission m is scheduled at go g
$vtmp_{m,p}$	$\mathbb{Q} \cap [0,1]$	ratio of training completion of mission m by pilot p
vtp_p	$\mathbb{Q} \cap [0,1]$	ratio of training completion by pilot p
$vtpu_p$	$\mathbb{Q} \cap [0,1]$	ratio of training completion by pilot p, who is following an upgrade training

$$\max \quad \mathcal{T} = \frac{w_r}{|\mathbf{P}_{QP}|} \sum_{p \in \mathbf{QP}} vtp_p + \frac{w_i}{|\mathbf{P}_{SP}|} \sum_{p \in \mathbf{IP}} vtp_p + \frac{w_u}{|\mathbf{P}_{UP}|} \sum_{p \in \mathbf{UP}} vtpu_p \tag{2a}$$

s.t.

$$\sum_{m \in \mathbf{M}'} \sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{m \in \mathbf{M}'_{UP}} \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} \leq na_w, \qquad \forall g \in \mathbf{G}', w = w', \tag{2b}$$

$$\sum_{m \in \mathbf{M}'} vs_{m,g,p} + \sum_{m \in \mathbf{M}'_{UP}} vsu_{m,g,p} \leq pa_{g,p}, \qquad \forall g \in \mathbf{G}', p \in \mathbf{P}, \tag{2c}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} = 4 \cdot vm_{m,g}, \qquad \forall m \in \mathbf{M}'_4, g \in \mathbf{G}', \tag{2d}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} = 2 \cdot vm_{m,g}, \qquad \forall m \in \mathbf{M}'_2, g \in \mathbf{G}', \tag{2e}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} = vm_{m,g}, \qquad \forall m \in \mathbf{M}'_1, g \in \mathbf{G}', \tag{2f}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} \leq 2 \sum_{p \in \mathbf{P}_{F2}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{2,RT}, g \in \mathbf{G}', \tag{2g}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} \leq 2 \sum_{p \in \mathbf{P}_{F2}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{4,RT}, g \in \mathbf{G}', \tag{2h}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} \leq 4 \sum_{p \in \mathbf{P}_{F4}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{4,RT}, g \in \mathbf{G}', \tag{2i}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} \leq 2 \sum_{p \in \mathbf{P}_{F4}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{4,U2}, g \in \mathbf{G}', \tag{2j}$$

$$\sum_{p \in \mathbf{P}} vs_{m,g,p} + \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} \leq \frac{4}{3} \sum_{p \in \mathbf{P}_{F2}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{4,U4}, g \in \mathbf{G}', \tag{2k}$$

$$\sum_{(m \in \mathbf{M}'_{BL} | m_{red} = n)} vm_{m,g} = vm_{n,g}, \qquad \forall n \in \mathbf{M}'_{RED}, \forall g \in \mathbf{G}', \tag{2l}$$

$$\sum_{p \in \mathbf{P}_{IP}} vs_{m,g,p} \geq \sum_{p \in \mathbf{P}_{SP}} vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{IL}, g \in \mathbf{G}', \tag{2m}$$

$$\sum_{m \notin \mathbf{M}'_{IL}} \sum_{g \in \mathbf{G}'} vs_{m,g,p} = 0, \qquad \forall p \in \mathbf{P}_{SP}, \tag{2n}$$

$$\sum_{g \in \mathbf{G}'} vs_{m,g,p} + c_{m,p} \leq r_{m,p}, \qquad \forall m \in \mathbf{M}'_{IL}, p \in \mathbf{P}_{SP}, \tag{2o}$$

$$c_{n,p} + \sum_{h=1}^{g-1} vs_{n,h,p} + vs_{m,h,p} \geq vs_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{IL,PRC}, g \in \mathbf{G}', p \in \mathbf{P}_{SP}, n \in \mathbf{PRC}_m, \tag{2p}$$

$$\sum_{p \in \mathbf{P}_{IP}} vs_{m,g,p} \geq \sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{UP}, g \in \mathbf{G}', \tag{2q}$$

$$\sum_{p \in \mathbf{P}_{UP}} vsu_{m,g,p} = vm_{m,g}, \qquad \forall m \in \mathbf{M}'_{UP}, g \in \mathbf{G}', \tag{2r}$$

$$\sum_{g \in \mathbf{G}'} vsu_{m,g,p} + c_{m,p} \leq r_{m,p}, \qquad \forall m \in \mathbf{M}'_{UP}, p \in \mathbf{P}_{UP}, \tag{2s}$$

$$c_{n,p} + \sum_{h=1}^{g-1} vsu_{n,h,p} + vsu_{m,h,p} \geq vsu_{m,g,p}, \qquad \forall m \in \mathbf{M}'_{UP,PRC}, g \in \mathbf{G}', p \in \mathbf{P}_{UP}, n \in \mathbf{PRC}_m, \tag{2t}$$

$$\sum_{m \in \mathbf{M}'_{U4}} \sum_{g \in \mathbf{G}'} \sum_{p \in \mathbf{P}_{U2}} vsu_{m,g,p} = 0 \quad , \tag{2u}$$

$$\sum_{m \in \mathbf{M}'_{U2}} \sum_{g \in \mathbf{G}'} \sum_{p \in \mathbf{P}_{U4}} vsu_{m,g,p} = 0 \quad , \tag{2v}$$

$$\sum_{m \in \mathbf{M}'} \sum_{g \in \mathbf{G}'} \sum_{p \in \mathbf{P}_{dy}} vs_{m,g,p} + \sum_{m \in \mathbf{M}'_{UP}} \sum_{g \in \mathbf{G}'} \sum_{p \in \mathbf{P}_{dy,UP}} vsu_{m,g,p} = 0 \text{ if } w' \in \mathbf{W}_{dy}, \qquad \forall dy \in \mathbf{DY}, \tag{2w}$$

$$\sum_{g \in \mathbf{G}'} vs_{m,g,p} + c_{m,p} \geq rdy_{m,p} \text{ if } w' = bw_{dy} - 1, \qquad \forall dy \in \mathbf{DY}, m \in \mathbf{M}'_{DY}, p \in \mathbf{P}_{dy}, \tag{2x}$$

$$vtmp_{m,p} = \left\{ \begin{array}{ll} max[0; \frac{r_{m,p} - c_{m,p}}{r_{m,p}}], & \text{if } \sum_{g \in \mathbf{G}'} vs_{m,g,p} + c_{m,p} \geq r_{m,p} \\ \sum_{g \in \mathbf{G}'} vs_{m,g,p}, & \text{else} \end{array} \right\} \text{if } r_{m,p} > 0, \quad \forall m \in \mathbf{M}'_{RT}, p \in \mathbf{P}_{QP}, \tag{2y}$$

$$vtp_p = \frac{\sum_{m \in \mathbf{M}'_{RT}} vtmp_{m,p} \cdot r_{m,p}}{\sum_{m \in \mathbf{M}_{RT}} r_{m,p}}, \qquad \forall p \in \mathbf{P}_{QP}, \tag{2z}$$

$$vtp_p = \frac{\sum_{g \in \mathbf{G}'} \sum_{m \in \mathbf{M}'_{IL}} vs_{m,g,p}}{\sum_{m \in \mathbf{M}_{IL}} r_{m,p}}, \qquad \forall p \in \mathbf{P}_{SP}, \tag{2aa}$$

$$vtpu_p = \frac{\sum_{g \in \mathbf{G}'} \sum_{m \in \mathbf{M}'_{UP}} vsu_{m,g,p}}{\sum_{m \in \mathbf{M}_{UP}} r_{m,p}}, \qquad \forall p \in \mathbf{P}_{UP}. \tag{2ab}$$

Objective Function The goal of each separate SPr is to maximize the average training value per pilot in each week, given the results of all preceding weeks. This goal is translated to the objective function in equation 2a as follows. The first term in the objective function represents recurrent training, the second term initial training and the third term transition training. Every term represents for that specific training type a ratio of how much of the training demand has been completed during the week, averaged over the number of pilots that are assigned that type of training. By formulating this as a ratio instead of absolute number of sorties, we can directly get an idea of how the training completion and ultimately the readiness increases every week. The weighting factors w_r, w_i and w_u are user-defined and can reflect the priority given to each of the training types.

Constraints The constraints can be divided into five categories.

First, there are constraints 2b and 2c which relate to resource availability. Constraints 2b state that for every go, the number of conducted sorties can not be higher than the amount of available aircraft, where the subscript UP indicates the subset that is concerned with transition training. Similarly, constraints 2c ensures that a pilot can only perform one sortie per go that the pilot is available.

Second, constraints 2d - 2l are constraints that enforce certain mission characteristics. Constraints 2d makes sure four pilots are assigned to a 4-ship mission ($\mathbf{M}'_4$), while constraints 2e and 2f do the same for 2-ship ($\mathbf{M}'_2$) and singleton ($\mathbf{M}'_1$) missions. Then, constraints 2g - 2k state that enough 2-FLs and 4-FLs are assigned to recurrent training missions ($\mathbf{M}'_{RT}$) of appropriate sizes. Due to the nature of the missions, upgrade missions

($\mathbf{M}'_{U2}$ or $\mathbf{M}'_{U4}$) require different amounts of flight leads than recurrent and initial training missions. This is a result of the fact that pilots in transition training perform a role in the formation which is not in accordance with their current qualification. Constraints 2l enforce that red air missions ($\mathbf{M}'_{RED}$) are planned simultaneously with the blue air missions ($\mathbf{M}'_{BL}$) that require opposing forces.

The next category of constraints, 2m - 2p, impose restrictions on initial training missions ($\mathbf{M}'_{IL}$). Constraints 2m ensure that an SP is always accompanied by an IP and constraints 2n states that SPs can only be assigned to initial training missions. Furthermore, those training missions can not be flown more often than dictated by the syllabus, which is specified by constraints 2o. Constraints 2p state that missions with precedents ($\mathbf{M}'_{PRC}$)can only be started when all preceding missions have been completed at least once.

Subsequently, there is the category that focuses on upgrade training in constraints 2q - 2v. Constraints 2q, 2s and 2t are similar to constraints 2m, 2o and 2p respectively, but deal with transition training instead of initial training. Additionally, Constraints 2r enforce that a transition training mission can only be planned if a pilot in upgrade training is assigned to it. Constraints 2u and 2v make sure that pilots in 2-FL training can not be assigned to 4-FL training and vice versa.

In the fourth category, constraints 2w and 2x handle the implementation of deployments, where constraints 2w state that a pilot can not complete training missions when on deployment and 2x are constraints that state that a pilot must complete the deployment work-up to be able to go on deployment.

Lastly, the constraints 2y - 2ab are introduced to calculate the actual training completion ratio. Constraints 2y are auxiliary, to calculate for each pilot the ratio of completion for each recurrent training mission. These constraints are required as every pilot could perform more executions of a single mission than required. However, these superfluous missions do not result in higher training value and as such can not be counted towards the objective. Therefore, constraints 2y checks for every pilot how many required executions are remaining for every mission. If a pilot performs any mission more times than was required, only the number of required executions at the beginning of the week counts towards the objective function. Then finally, constraints 2z, 2aa and 2ab compute the incremental training completion ratio respectively for recurrent, initial and transition training in this SPr. These values are then transferred over to the objective function.

4 Case Study

To evaluate the performance of the proposed FPTSAP model, we test the model with a case study. For the case study, two closely collaborating squadrons operating F-16 fighter jets out of RNLAF Volkel airbase are considered. This case is outlined further in Subsection 4.1. Unfortunately, no historical data was available within the RNLAF to compare the performance of the proposed model to. As an alternative, a rule-based decision making algorithm was developed which is introduced in Subsection 4.2.

4.1 Case Description

Both 312 and 313 squadrons are located at Volkel airbase and currently solely operate F-16 fighter jets. While they are both operational squadrons, they are also tasked with the (partial) fulfillment of initial, recurrent and transition training. Each pilot in the squadrons has one of the following qualifications: IP, 4-FL, 2-FL, WM or SP. SPs are new to the squadron and have to complete initial training. All qualifications other than SP combined are classified as QPs and have to complete recurrent training. Some more experienced 2-FL and WM can be assigned to follow transition training, on top of the recurrent training, to become 4-FL and 2-FL, respectively. On top of the qualification, every pilot has a status of either experienced or inexperienced which further influences the number of required completions per training mission.

Depending on pilot experience and the assigned syllabi, every pilot should execute every training mission a certain amount of times. Every pilot then ideally should complete the training missions while using the fewest amount of weeks possible. The full scheduling horizon is 52 weeks, but one would not want to use all these weeks for the baseline training schedule, as has been explained before. Continuing, there are five working days in a week and every day is composed of two goes, AM and PM. Every week either four, six or eight aircraft are available for scheduling at each go. Every pilot is also assigned some days off in the schedule, which can be a result of numerous reasons for planned employee absence.

The total reference scenario consists of 23 pilots, 56 unique blue air missions and three red air missions. Lastly, for the reference scenario τ from Table 5 is set to 0.95, which indicates that the FPTSAP aims to have all pilots complete at least 95% of their training programs. We set this value at 0.95 to prevent the model from spending excessive time on realising a marginal increase in readiness.

There is also an option to include deployments in the schedule. For the scenario, a deployment can be included which is split into two periods of four consecutive weeks. The first period spans from week 6 up to and including week 9 and the second period spans from week 10 up to and including week 13. For every deployment four pilots should be appointed. Combined, those pilots should have the required qualifications to be able to form a 4-ship formation. Besides the pilots, aircraft should also be made available for the deployment. To include this in the scenario, the number of aircraft available for training during the deployment periods is fixed at four.

4.2 Rule Based Algorithm

A Rule-Based Algorithm (RBA) has been developed to simulate the current manual scheduling process at the mentioned squadrons. At the beginning of the year, the decision is made to plan every mission category one after another in a repeating manner. If there were five categories, the weekly planning would just be to schedule category 1 first, then 2, then 3 and so on. Then after category 5, category 1 is scheduled again. Before the start of the upcoming week, the scheduler greedily constructs a schedule per go from the available pilots and aircraft. Priority is first with transition training, then initial training and lastly recurrent training. Subsequently, the scheduler prioritizes larger missions over smaller missions and tries to assign the pilots with the most executions remaining to these missions first. At the end of each week, the scheduler checks if all pilots have completed all required missions for the scheduled mission category. If this is the case, this mission category is removed from the schedule and all remaining categories are moved forward in the schedule to fill up resulting gaps. This terminates the week and a new week begins, where the scheduler repeats this process of building a schedule per go. This process is captured in a RBA, for which the pseudo-code is given in Algorithm 2 in Appendix C.

5 Results

We must first identify how many iterations of the solution process should be made. The Constraints in the MPr effectively reduce the solution space for a problem the size of the reference scenario. Therefore, the expectation is that only a small number of iterations is needed to meet the training objective specified by τ. A series of tests is carried out to investigate the performance for the reference scenario. Per test, 25 runs are executed. A run is defined as a single completion of the process visualized in Figure 1, where a new problem instance is created every run. The differentiation between the instances comes from the availability of aircraft per week and the availability of pilots per day, as these are the input parameters that are beyond our control. The results are visualized in Figure 3.

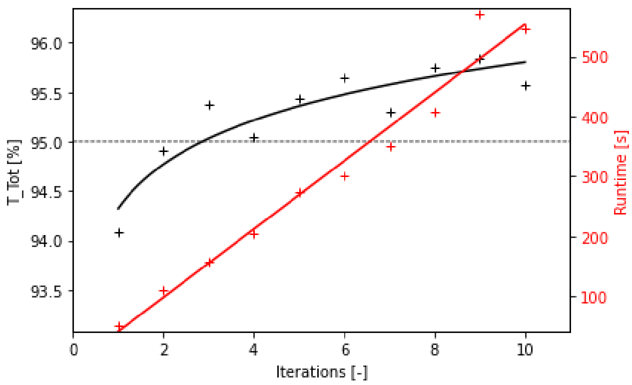


Figure 3: Average total training completion and model runtime for 25 runs and for varying iterations, with $\tau = 0.95$.

Every iteration solves all the SPrs again for the same problem instance but a different MPr solution. Only minor computational improvements can be gained with the MIP starts that are implemented in the SPrs. As such, solving every iteration is of equal complexity and the linear increase in runtime per iteration is as expected. At the same time, it can be seen that the total training completion passes the objective of τ at just over three iterations. Therefore, for the remainder of this paper the number of iterations used is set to four. At this number of iterations, the desired training value denoted by τ can be met in the smallest amount of time. However, one could also choose to let the model run for some more iterations, as the total training value still gradually increases.

Further performance of the FPTSAP in the reference scenario is benchmarked against the RBA in Section 5.1. To investigate how the FPTSAP model reacts to problems of different size and complexity a sensitivity analysis is presented in Subsection 5.2.

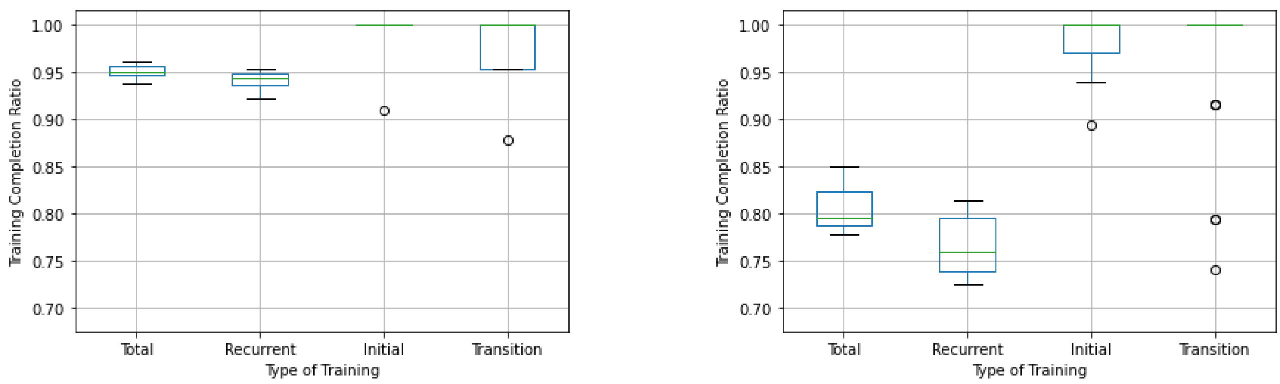
5.1 Benchmark

To get an idea of how the FPTSAP model performs with regard to the current situation of manual scheduling, the FPTSAP model is benchmarked against the RBA. As the RBA is not able to deal with deployments, during the benchmark the FPTSAP model also disregards deployments. Furthermore, the RBA schedules transition training first, then initial training and lastly recurrent training. To implement this in the FPTSAP, w_r, w_i and w_u are set to 1, 100 and 1000 respectively. These values are chosen arbitrarily to reflect the same strictness in the priorities as the RBA uses. To be able to come to a decent comparison, 25 runs are executed for both the RBA and the FPTSAP model.

Running the FPTSAP model for the reference scenario, most solutions use 23 weeks to fulfill the training demand and construct the schedule. As such, 23 weeks are made available for the RBA as well. An overview of the results of these runs is given with boxplots in Figure 4. Furthermore, the averages of the training completion ratios and sorties over 25 runs are given in Table 10. It can be concluded that the FPTSAP model leads to an increase of total training completion of over 15 percentage points with respect to the RBA. The FPTSAP model manages to deliver such an increase mainly by using approximately 12 percentage points more sorties than the RBA. This is a logical consequence of the fact that the FPTSAP model is an optimization model that seeks the most efficient allocation of resources, while the RBA uses a greedy approach and as such regularly leaves multiple aircraft unused. Still, the 84.42% of sorties used by the FPTSAP model ideally should be higher. Inspecting the constructed schedules, it becomes apparent that weeks can occur in which less than 10% of the available sorties are used. This is a consequence of formulation of the MPr objective and decision variables. The MPr can only schedule mission categories for full weeks, which is not always consistent with the demand for sorties in those mission categories. This leads to inefficiently scheduled weeks and scheduling opportunities that are left unused.

Analyzing the different training types, it can be seen that the increase in average training completed comes almost solely from an increase in recurrent training completion, where the FPTSAP model scores better than the RBA. For completion of initial training, the average of the FPTSAP model is about one percentage point higher than that of the RBA. The average completion of transition training is roughly two percentage points higher for the FPTSAP, but this is mainly due to some negative extremities in the RBA. Eliminating these extremities, the RBA does a better job at adhering to the transition training priority. That being said, with an increase of 15 percentage points in total training completion it can be concluded that for the reference scenario, the FPTSAP model outperforms the RBA and as such outperforms the current scheduling process. Additionally, Table 10 also presents the average percentage of pilots that have completed all the training (of a specific type) that was assigned to them. A remarkable metric is that the RBA does not succeed to have any pilot fully complete the recurrent training syllabus.

Lastly, the runtime for a single run of the RBA averages to three seconds and that of the FPTSAP model to 205 seconds. While this is a hefty increase, the FPTSAP still comes up with a solution for a problem instance of real-life scale in less than four minutes, which is easily within operational requirements. Additionally, with a runtime this low the model can not only be implemented at the beginning of the year, but also if changes occur throughout the year. Then, one could simply modify the input data in accordance with the changes and construct a new schedule.



(a) FPTSAP model.

(b) RBA.

Figure 4: Boxplots of training completion ratios for both models, for 25 runs.

Table 10: Averages of training completion ratios and sorties over 25 runs, for the FPTSAP and RBA.

	FPTSAP	**RBA**
Percentage of Training Completed		
Total Training	95.05	80.42
Recurrent Training	94.16	76.42
Initial Training	99.64	98.55
Transition Training	98.18	96.30
Percentage of Full Syllabus Completions		
Total	70.46	14.62
Recurrent Training	67.05	0
Initial Training	96.00	72.00
Transition Training	77.33	78.67
Sorties	1165	1004
Sorties used	84.42 %	72.75 %

5.2 Sensitivity Analysis

To further research how the FPTSAP model performs under different circumstances and to be sure the performance from Section 5.1 is not a coincidence, some more tests are run. All results in this subsection are averages of 25 runs and are obtained with the training threshold value τ set to 0.95, unless specified otherwise. Scenarios are tested to research the influence of the size of the problem and the value of threshold parameter τ. Outcome parameters of interest remain the runtimes, number of weeks used, training completion ratios and number of sorties.

Problem Size There are two main aspects we can vary that influence the problem size: the number of pilots and the amount of missions they have to execute. For these tests, the number of pilots is varied between 10, 23 (reference scenario), 50 and 75. The number of sorties required is varied between the amount in the reference scenario and twice the amount in the reference scenario. The results are given in Table 11. It can be seen that even for the largest scenario, the computation time stays within 10 minutes. However, the total training completion ratio is below the threshold τ. As the scenario size increases, the completion ratio falls further below the specified value of τ of 95%. Additionally, we see that the model also has some difficulties with handling smaller problem sizes. This could logically be the result of a smaller solution space, so that resources can not always be allocated efficiently. We also ran some preliminary tests to look into the effect of increasing the amount of aircraft available per go. For the reference scenario, doubling the amount of aircraft available results in an increase in runtime of roughly 50% and a reduction in required weeks by 50%, while the amount of training completed and percentage of sorties used remains the same.

Table 11: FPTSAP model results for varying problem sizes.

#pilots	*sorties required*	**d** [s]	**w**	$\mathbf{T_{Tot}}$ [%]	$\mathbf{T_R}$ [%]	$\mathbf{T_I}$ [%]	$\mathbf{T_T}$ [%]	**sorties used [%]**
10	*Ref*	136	13	93.47	92.11	98.42	97.10	68.33
Ref	*Ref*	205	23	95.05	94.16	99.64	98.18	84.40
50	*Ref*	244	45	90.92	88.96	95.37	98.25	89.73
75	*Ref*	305	65	89.26	87.37	93.00	98.08	88.61
10	*2 ∗ Ref*	91	20	90.66	88.81	95.58	96.54	81.20
Ref	*2 ∗ Ref*	195	41	91.22	89.67	97.36	97.89	88.76
50	*2 ∗ Ref*	263	88	88.67	86.33	90.93	99.06	88.53
75	*2 ∗ Ref*	434	127	87.27	85.41	89.12	97.26	88.33

Note: d is runtime; w is number of weeks used; T stands for training completion, with subscripts R, I and T for recurrent, initial and transition training respectively.

Threshold Value τ is the training threshold value. If for instance it is set to 0.95, the MPr should theoretically construct a weekly schedule that enables the SPrs to come up with a solution that has a total training completion of 95%. In this Section, τ is adjusted between 80% and 100%. This is done for the reference scenario. Ideally, total training completion should follow the value of τ. It can be concluded from Figure 5 that it only does when τ is set to 95%. For lower values of τ, the inaccuracies in total completed training are a result of the MPr formulation. The MPr schedules the minimum integer amount of weeks per mission category that is required to fulfill the mission demand specified by τ for that mission category. However, for some mission categories the amount of weeks required to meet a τ of 80% might be the same as the amount of weeks required to meet a τ of 82.5% and and so forth. Therefore, low values of τ can result in higher than required values of total training completed. The inaccuracies in total training completion for values of τ higher than 95% can be attributed to inefficiencies in the SPrs and the RH approach that does not use a prediction horizon. While this optimal value of 95% is acceptable, this value is specific to our test case. For other cases, the value where total training completion meets τ might be lower and unacceptable.

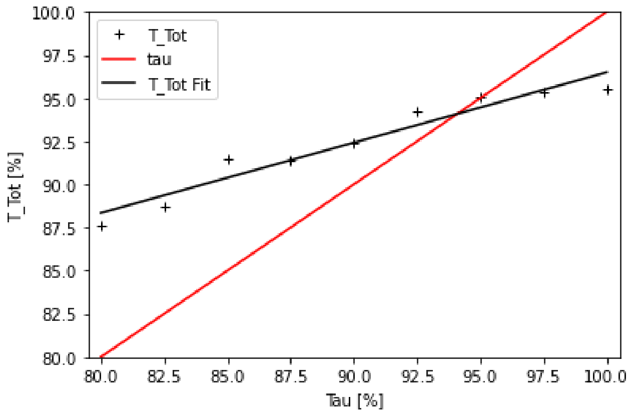


Figure 5: Average total completed training for varying τ.

5.3 Deployments

One advantage that the FPTSAP model has over the RBA, is that it can factor in deployments and adequately prepare those pilots that are assigned to the deployment. This has impact on the schedule, because the deployment pilots might need to perform extra missions and because they are taken out of the schedule during their deployment. It is investigated here to what extent this could influence their ability to fulfill their training requirements. The pilots that are selected to go on a deployment, are chosen before constructing the preliminary schedule. In the current model, the pilots are chosen arbitrarily and the only requirement is that they should be able to form a 4-ship formation. More requirements on the choice of pilots that go on deployment could easily be defined if the user demands this. The selected pilots have to complete the deployment work-up program before the start of the deployment. In some iterations of the FPTSAP model, this leads to an infeasibility in one of the SPrs. The infeasibilities come from the solution to the MPr, which does not always provide the SPrs with the solution space required to have the selected pilots complete the work-up program in time. Still, since we perform 4 iterations per run, a feasible solution to the overall problem is output every run.

Including the deployment in the scenario leads to an average total training completion ratio of 94.28%, a decrease of 0.77 percentage points compared to the scenario without deployment. This decrease primarily is the result from a decrease in transition training completed. The mission category planning per week typically reflects the precedent relations between the missions in those mission categories. A pilot who is sent on deployment can not perform training missions during his deployment. During that period, the pilot can miss out on weeks that provide opportunities to perform a mission which is a critical precedent to other missions. Then if the pilot returns from his deployment, it is harder to continue the transition training syllabus, because the pilot does not meet the precedent requirements for the missions that can be scheduled in the following weeks. This results in lower average completion of transition training. This problem only occurs for transition training, as recurrent training missions do not have precedents and SPs (in initial training) can not go on deployments. As such, we can conclude that for this scenario, it is possible to implement a deployment in the schedule which only leads to a minor decrease in total training completed.

6 Conclusion

In this paper a model is proposed that schedules and assigns fighter pilot training missions. The ultimate goal of this model is to increase the personnel readiness of fighter pilots in the RNLAF. Previously proposed solutions to the FPTSAP often focused on scheduling either initial training or recurrent training and did not include transition training at all. Our model is capable of scheduling initial training, recurrent training and transition training. Additionally, an option is included in the model that enables pilots to train for and go on deployments. The model was originally designed to be able to schedule for a full year, starting at the beginning of the calendar year. However, with a few alterations, it could be used to maximize the amount of training for any horizon length, with any starting situation and for a variety of weapon systems.

To solve the FPTSAP, we split the overarching problem into two stages both formulated as a MIP model. This choice was primarily made for computational efficiency. An additional benefit that comes from this two-stage approach is that the different schedules, which have different levels of detail, can be used during different stages of the planning process and on different organizational levels. During the first stage, the MPr, a preliminary schedule is created. This schedule determines which mission category is scheduled for each week in the scheduling horizon. Subsequently in the second stage, a detailed schedule is constructed for every week by the SPrs using a RH approach.

The model was tested on a case that studied two RNLAF fighter squadrons and compared to a RBA that resembles the current scheduling process at the RNLAF. The metric that represents readiness of personnel in the model is the amount of training completed averaged over all the pilots. For this metric, the proposed model results in an increase of more than 15 percentage points when compared to the RBA. This leads to an average of 95% of training completed per pilot, which is consistent with the requirements set by the threshold value. At the same time, in this particular case the FPTSAP model succeeds to have 67% of all the pilots complete their full recurrent training syllabus compared to zero pilots completing all their training in the RBA solution. Our FPTSAP model solves the reference scenario within 4 minutes. This runtime easily allows for new schedules to be built throughout the year when changes occur in training demand or resource availability. Furthermore, we implemented the option to include a deployment into the schedule at the cost of a marginal decrease of one percentage point in average total training completion.

On the other hand, on average the model only uses 85% of the sorties available for scheduling. This is mainly a consequence of formulation of the MPr, which lacks flexibility in scheduling options. Moreover, for larger problem instances the average of total training completed is below the desired threshold. This is also attributable to the MPr formulation. For larger problem instances, the MPr can have a large amount of solutions with the same objective value. The cuts that are used at each iteration fail to decrease the solution space fast enough to be able to come to a good quality solution within the number of iterations specified.

We can therefore conclude that for our reference scenario, the proposed model succeeds to increase the readiness of personnel to the desired value. However, improvements can be made in terms of schedule efficiency and scalability. A possible starting point to realise these improvements would be to reformulate the MPr objective in such a way that every solution produces its own unique value for the objective function. Further improvements could also be made by increasing the efficiency of the cuts that are implemented after each iteration of solving the overarching problem or by introducing more flexibility in scheduling the mission categories. Additionally, there is also room for improvement in the formulation and solution process of the SPrs. The RH currently has no prediction horizon and only takes into account past weeks and the current weeks. However, inclusion of a prediction horizon could be beneficial to prevent infeasibilities in some cases where hard deadlines are set, such as when a pilot has to complete a deployment work-up program before the start of the deployment. For this reason, future research could be dedicated to implementing a prediction horizon in the model.

Lastly, the semantics, sets and subsets in the model as well as some assumptions have their origin in the RNLAF fighter branch. However, the model was developed with the aim to be applicable to other Air Forces as well. With some minor adjustments to the model, this should still be the case. Besides, interest into a similar tool is also expressed by the rotary aviation sector of the RNLAF. To those ends, future research could be dedicated to test the model in other Air Forces and other branches of military aviation.

References

[1] Davut Aslan. A decision support system for effective scheduling in an f-16 pilot training squadron. Master's thesis, Air Force Institute of Technology, 2003.

[2] G.E.P. Box. *Robustness in statistics*, chapter Robustness in the strategy of of scientific model building, pages 201–236. Academic Press, 1979.

[3] Jens Clausen. Brand and bound algorithms: Principles and examples. techreport, University of Copenhagen, 1999.

[4] Ugur Erdemir. Optimizing flight schedule by an automated decision support system. Master's thesis, Air Force Institute Of Technology, 2014.

[5] Royal Netherlands Air Force. Clsk/ops/v-40/f-16 trainingsprogramma, 2018.

[6] Royal Netherlands Air Force. Clsk/ops/v-22/f-16 mission qualification training, 2020.

[7] R. S. Garfinkel and G. L. Nemhouser. The set-partioning problem: Set covering with equality constraints. *Operations Research*, 17(5):848–856, 1969.

[8] Reinout Guljé. A priority-based method for scheduling and rescheduling fighter pilot continuation training. Master's thesis, Erasmus University Rotterdam, 2018.

[9] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[10] Asa Holm. Manpower planning in airlines. Master's thesis, Linköpings Universitet, 2008.

[11] Bong-Kyun Kim and Yeong-Dae Kim. Heuristic algorithms for assigning and scheduling flight missions in a military aviation unit. *Computers & Industrial Engineering*, 61:1309–1317, 2011.

[12] George Kozanidis. Optimal assignment of aircrew trainees to simulator and classroom training sessions subject to seniority and preference restrictions. *Journal of Air Transport Management*, 59:143–154, 2017.

[13] Thomas M. Newlon. Mathematical programming model for fighter training squadron pilot scheduling. Master's thesis, Air Force Institue of Technology, 2007.

[14] Cuong T. Nguyen. An interactive decision support system for scheduling fighter pilot training. Master's thesis, Air Force Institute of Technology, 2002.

[15] DuyQuang Nguyen and Miguel Bagajewicz. Optimization of preventive maintenance scheduling in processing plants. *Computer Aided Chemical Engineering*, 25:319–324, 2008.

[16] Office of the Chairman of the Joint Chiefs of Staff. *DOD Dictionary of Military and Associated Terms*, page 62. Washington DC: The Joint Staff, 2021.

[17] Michael L. Pinedo. *Scheduling. Theory, algorithms and systems.* Springer, 2016.

[18] Xiangton Qi, Jonathan F. Bard, and Gang Ju. Class scheduling for pilot training. *Operations Research*, 52(1):148–162, January 2004.

[19] Mauricio G. C. Resende and Celso C. Ribeiro. *Handbook of metaheuristics*, chapter Greedy randomized adaptive search procedures, pages 219–249. Kluwer Academic Publishers, 2003.

[20] G. T. Ross and R. M. Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming*, 8:91–103, December 1975.

[21] S. Salcedo-Sanz. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Physics Reports*, 655:1–70, 2016.

[22] Javier Silvente, Georgios M. Kopanos, Efstratios N. Pistikopoulos, and Antonio Espuña. A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids. *Applied Energy*, 155:485–501, 2015.

[23] Mike Taks. Scheduling training missions and pilots. Master's thesis, Utrecht University, 2016.

[24] Dirk Jan van den Berg, Michael Beltman, Bas Bijlsma, Maarten Boef, Henk Geveke, Stan Kaatee, Niels Kastelein, Gijs van Keulen, Pieter van Marken, Peter Reesink, Philip Strik, Gerbe Verhaaf, Arthur van Vliet, David van Weel, and Hans Wehren. Zicht op gereedheid. Technical report, Rijksoverheid, 2017.

[25] Joey van Kempen. Robust cockpit crew training scheduling. Master's thesis, Delft University of Technology, 2019.

[26] Murat Yavuz. Optimizing an f-16 weekly pilot schedule for the turkish air force. Master's thesis, Air Force Institute of Technology, 2010.

[27] Gang Yu, Julian Pachon, Benjamin Thengvall, Darryal Chandler, and Al Wilson. Optimizing pilot planning and training for continental airlines. *INFORMS Journal on Applied Analytics*, 34(4):253–264, July 2004.

Appendices

A Appendix 1

A partial example of a schedule constructed by the FPTSAP model is given in Table 12. The table visualizes the planning per go. In this schedule, we can see that pilot *16-F2-4* is assigned to mission 1 during the AM go of day 1. In the scenario used, a maximum of eight aircraft is available per week. If less than eight aircraft are available for a given week, the schedule is completed with *na*. If an aircraft was available, but not used for scheduling, the schedule displays *Empty AC*.

Table 12: Partial example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM1	1: 16-F2-4	1: 22-SP-1	1: 23-SP-2	1: 2-IP-2	1: 7-IP-7	1: 8-F4-1	2: 18-F2-6	2: 19-WM-1
PM1	1: 5-IP-5	1: 6-IP-6	2: 22-SP-1	2: 23-SP-2	2: 2-IP-2	2: 3-IP-3	3: 18-F2-6	3: 19-WM-1
AM2	1: 13-F2-1	1: 1-IP-1	2: 12-F4-5	2: 14-F2-2	38: 19-WM-1	38: 20-WM-2	38: 2-IP-2	38: 7-IP-7
PM2	1: 15-F2-3	1: 9-F4-2	35: 23-SP-2	35: 7-IP-7	39: 20-WM-2	39: 5-IP-5	4: 10-F4-3	4: 1-IP-1
AM3	2: 4-IP-4	2: 7-IP-7	2: 8-F4-1	2: 9-F4-2	3: 1-IP-1	3: 22-SP-1	3: 23-SP-2	3: 3-IP-3
PM3	2: 20-WM-2	2: 5-IP-5	35: 1-IP-1	35: 23-SP-2	39: 19-WM-1	39: 3-IP-3	4: 22-SP-1	4: 7-IP-7
AM4	1: 10-F4-3	1: 11-F4-4	1: 12-F4-5	1: 14-F2-2	1: 20-WM-2	1: 4-IP-4	2: 16-F2-4	2: 21-WM-3
PM4	1: 18-F2-6	1: 3-IP-3	2: 10-F4-3	2: 11-F4-4	2: 13-F2-1	2: 6-IP-6	3: 16-F2-4	3: 21-WM-3
AM5	35: 11-F4-4	35: 19-WM-1	3: 17-F2-5	3: 20-WM-2	4: 23-SP-2	4: 6-IP-6	5: 1-IP-1	5: 22-SP-1
PM5	1: 17-F2-5	1: 21-WM-3	2: 15-F2-3	2: 1-IP-1	35: 6-IP-6	5: 23-SP-2	5: 2-IP-2	Empty AC
					...			
AM121	36: 10-F4-3	36: 13-F2-1	51: 12-F4-5	51: 19-WM-1	51: 1-IP-1	51: 6-IP-6	na	na
PM121	29: 16-F2-4	29: 20-WM-2	36: 12-F4-5	36: 13-F2-1	Empty AC	Empty AC	na	na
AM122	29: 18-F2-6	29: 7-IP-7	36: 11-F4-4	36: 15-F2-3	Empty AC	Empty AC	na	na
PM122	29: 10-F4-3	29: 16-F2-4	36: 5-IP-5	36: 6-IP-6	Empty AC	Empty AC	na	na
AM123	29: 17-F2-5	29: 21-WM-3	36: 12-F4-5	36: 22-SP-1	Empty AC	Empty AC	na	na
PM123	36: 17-F2-5	36: 1-IP-1	51: 15-F2-3	51: 20-WM-2	51: 4-IP-4	51: 7-IP-7	na	na
AM124	29: 16-F2-4	29: 20-WM-2	36: 14-F2-2	36: 4-IP-4	Empty AC	Empty AC	na	na
PM124	29: 17-F2-5	29: 7-IP-7	36: 15-F2-3	36: 22-SP-1	Empty AC	Empty AC	na	na
AM125	29: 18-F2-6	29: 20-WM-2	36: 15-F2-3	36: 22-SP-1	Empty AC	Empty AC	na	na
PM125	29: 17-F2-5	29: 21-WM-3	36: 20-WM-2	36: 2-IP-2	Empty AC	Empty AC	na	na

B Appendix 2

This Appendix provides pseudo-code that describes the high-level solution process of the proposed FPTSAP model.

Algorithm 1 High level solution process of the FPTSAP model

```
1: Input Pilot Data, Mission Data, number of iterations I
2: for i to I do
3:   Solve MP: minimize required weeks, construct mission category schedule
4:   With the MP solution:
5:   for w to W do
6:     Solve SP: maximize training value in current week
7:     Update completed sorties
8:     Continue to next week
9:   end for
10:  Save total training value
11:  if Total training value exceeds stopping criteria then
12:    Proceed to step 18
13:  else
14:    Implement cuts
15:    Continue to next iteration
16:  end if
17: end for
    // ∗ denotes best iteration
18: Select best solution from iterations
19: return Minimum number of weeks, maximum training value
20: Output Schedules, Training Completion Values
```

C Appendix 3

This Appendix provides pseudo-code that describes the RBA that reflects the current scheduling process and that is used to benchmark the FPTSAP model.

Algorithm 2 Pseudo-code of the RBA

```
Input Pilot Data, Mission Data, number of weeks W
// mc is mission category, start at first in list
for w to W do
   Schedule mc for week w
   Move to next mc in list
   if mc = last mc in list then
      Restart at first mc
   end if
end for
for w to W do
   With mc = scheduled mc in week w
   for go in w do
      for s in [transition, initial, recurrent] do
         if pilots & aircraft available, executions required then
            Plan missions, larger missions first
            Update pilot & aircraft availability
            Update completed executions
         else
            Continue to next syllabus
         end if
      end for
      if All missions in mc executed then
         Remove mc from future schedule
      end if
   end for
end for
Output Training schedules, training completion values
```

II

Literature Study

Previously graded under AE4020

1 Introduction

It is a well accepted point of view that being a fighter pilot belongs to the most demanding jobs in the world. In order to become a fighter pilot, a long lasting and intensive initial training is required. But even after pilots have completed the initial training, much of their yearly flying program consists of sortie with the aim of maintaining their level of proficiency. As the RNLAF resides among the best trained and most advanced air forces in the world, their fighter pilot training programs are equally ambitious. While the Lockheed Martin F-35 is now being introduced in the RNLAF, the RNLAF currently operates the General Dynamic (now Lockheed Martin) F-16 'Fighting Falcon' as their only Fully Operationally Capable (FOC) fighter aircraft. The F-16 is designed and operated as a day-and-night all weather multi-purpose fighter aircraft. As a result, Dutch F-16 pilots have to be trained to be proficient in carrying out all sorts of tasks in any weather condition.

To gain and maintain the RNLAF's pilots level of proficiency, several training syllabi, such as the Mission Qualification Training (MQT) and the OPS V-40 have been composed. These documents describe, among other things, a set of specific training missions that have to be flown by each pilot so that he can achieve or maintain his currency. While scheduling these sorties, the available resources as well as the imposed constraints have to be taken into account. For example, different pilot qualifications, airframe and crew availability and different mission types have to be considered. Furthermore, fighter pilot operations are typically carried out by formations of jets rather than individually and as such, so are the training missions. Consequently, completion of pilots' training programs depend on the availability of other pilots. All in all, it is apparent that the planning process is a heavily constrained and complex environment. Nevertheless, planning is currently done manually as a secondary task by (ex-)fighter pilots who do not have scheduling as their main area of expertise. As such, planning is regularly done ad-hoc or on a very short notice with few, if any decision support tools available to the schedulers Consequently, scheduling is a time consuming task and now results in sub-optimal and even infeasible schedules. The research succeeding this literature survey aims to provide a planning tool that assists schedulers in forming a feasible and effective yearly schedule, so that all pilots can fulfill their training requirements.

To achieve this goal, it is insightful to explore the research that has already been done on fighter pilot scheduling and closely related subjects. The aim is to identify where the succeeding research fits in the existing literature, where the research gap lies and how the existing literature can provide useful insights for the research to follow.

The structure of this report is as follows. After this introduction, the research context the problem fits into is investigated in Chapter 2. The methodologies which are frequently used in scheduling problems and which have been applied in aircrew training scheduling literature are discussed in Chapter 3. Lastly, Chapter 4 determines how our research can contribute to the existing literature and how this contribution can be realized.

2

Defining the research context

In this chapter, we explore existing literature that focuses on the scheduling and assignment of aircrew training. By doing so, we aim to identify the current level of understanding of aircrew training scheduling and assignment, how several works interrelate, the advantages and shortcomings and recommendations for future research. As the problem of scheduling and assigning aircrew training in military aviation is vastly different from that in civil aviation, those two fields of research are split up over two sections. Section 2.1 outlines literature that focuses on scheduling problems that relate to military aviation, more specifically for fighter pilots. The majority of this research was done at the Air Force institute of Technology (AFIT). Subsequently, Section 2.2 focuses on training scheduling problems that have been researched in the civil aviation sector. The last section, Section 2.3 combines the findings from the previous sections to define the research gap in scheduling and assigning fighter pilot training.

2.1. Scheduling training in military aviation

While extensive research exists on all kinds of scheduling problems within aviation and other fields of work, the military scheduling environment is so different from the civil environment that this research can only partially be applied to military studies. As scheduling within military aviation units often is still done manually however, research dedicated to optimizing military aircrew scheduling could result in great improvements in efficiency and costs. Some research already exists that focuses on the FPTSAP. This existing literature can be divided into two fields of research; scheduling initial training and scheduling recurrent training.

2.1.1. Scheduling fighter pilot initial training

There have been a few studies on fighter pilot scheduling, the earliest to be discussed is the research paper by Nguyen [17]. Nguyen attempts to build a decision support tool to support planners at a United States Air Force (USAF) fighter training squadron that provides initial training. He states that the FPTSAP is complex. Despite the complexity, scheduling and assignment at USAF squadrons is often done manually by experienced fighter pilots who do not have scheduling as their expertise. Therefore, he develops a model that supports the planner in scheduling both classroom sessions for groups of students as well as assigning individual sorties and simulator sessions. The main organizational objective of such a tool is to decrease the time spent on constructing feasible schedules drastically. The training squadron Nguyen focuses on has to provide multiple classes of fighter pilot trainees with a 120-day training program. The syllabus that is to be completed is fixed, but at any time different classes within the squadron are at a different point in the training syllabus. Consequently, each class requests different training requirements. At the end of each week, a weekly schedule is developed for the next week, while each day is split up into three goes. This schedule has to be constructed in such a way that students can complete their training syllabus within 120 days. Nguyen also recognizes that the environment is prone to many dynamic aspects, such as aircraft maintenance and weather, and tries to implement this in the decision support tool as well. He simulates these effects with a so-called attrition model that renders some missions unsuccessful based on a predetermined fixed probability. A basic representation of how Nguyen sees the scheduling environment is given in Figure 2.1. In order to solve the problem mentioned above, Nguyen uses a Visual Interactive Modelling (VIM) approach and a Construction Heuristic (CH) which makes its choices based on several dispatching rules (3.2.2). The main

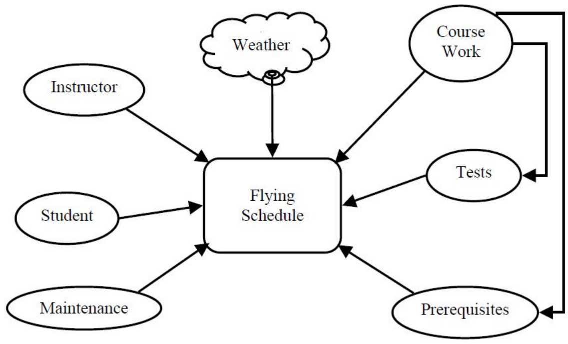


Figure 2.1: Basic representation of Nguyen's scheduling problem environment [17].

objective of the model is to complete the training syllabus within 120 days, while maximizing sorties and also adhering to the classroom and simulator sessions requirements. Nguyen concludes that feasible schedules with sorties above the minimum required amounts can be developed in reasonable time on a decision support tool that is easy to use for the squadron schedulers. For future research, he recommends to expand the attrition model he used and to further explore the usage of dispatching rules [17].

In many aspects, the research done by Aslan is similar to that of Nguyen [1]. Both works focus on providing a decision support tool to assist schedulers in scheduling and assignment of fighter pilot initial training. In Aslan's research, much like in Nguyen's, the syllabus and sequence within the syllabus that each student has to complete is known. On top of that, the Turkish Air Force (TUAF) squadron Aslan's research is based on also has operational mission demand which influences instructor availability. To plan a sortie, a single student and a single instructor have to be planned simultaneously. The research does not take into account that the availability of fighter jets is limited. The goal of the research is to develop a model which is able to reduce the time in which, contrary to Nguyen's weekly schedule, a daily schedule can be found. Aslan model splits the problem into three sub-problems in accordance with the pilots' qualifications and solves each of these sub-problems with a CH. During the construction of such a schedule, additional objectives are to balance the workload over the instructors and to balance the training sorties over the students. The results of this research are promising: Aslan manages to decrease the duration of the standard training program by about 20 percent. He does however make an important note that more thought could be put in to resource availability and influences such as weather and pilot absence [1].

2.1.2. Scheduling recurrent fighter pilot training

Both Nguyen and Aslan have researched scheduling problems in fighter pilot initial training. As a result, training requirements are fixed and sequentially ordered according to a predetermined syllabus. Scheduling recurrent training for pilots in (operational) fighter squadrons is more complicated due to the nature of the training and complexity of the training missions. There are two main differences between recurrent training of readily qualified fighter pilots and initial training of trainee pilots. The first one lies in the syllabi. During initial training, the missions within the syllabus are followed in a strictly sequentially order. During recurrent training however, precedence relations might exist, but they are often not as strict as during initial training. The second difference is that fighter pilot trainees often only fly alone or in simple formations, while qualified fighter pilots have to perform missions in larger formations and also rely on supporting aircraft. As a result, completion of single pilot training programs is dependent on the availability of other pilots. These pilots in turn have their own training requirements. For example, one could be in a program to become an instructor or to upgrade his qualification to be allowed to lead larger flights, both of which add extra complexity.

Research on such scheduling problems was carried out by Newlon, Yavuz, Erdemir, Taks and Guljé ([16], [30], [4], [26], [9]). Newlon's research is a scheduling and assignment problem, with a scheduling horizon of one week, based on an USAF squadron [16]. The main goal of the operations within this squadron is focused on maintaining pilot currencies in accordance with different criteria. Therefore, all pilots have to undergo a recurrent training syllabus. Every day in the week is divided into two go in which sorties can take place. The schedule has to consider planning flight sorties, simulator flights and ground events to be carried out by pilots. In planning the sorties, pilot qualifications have to be taken into account. Newlon differentiates between three pilot qualifications: Instructor Pilot (IP), Flight Lead (FL) and Wingman (WM). At the same time however, Newlon assumes that there is a freely available and unlimited amount of reserve pilots which can be used to fill gaps in the schedule. Similar to Nguyen and Aslan earlier, Newlon's goal is to find a feasible schedule in reduced time. Newlon formulates the problem as a Mixed Integer Programming (MIP) problem, which can either be solved for a whole week or be divided into ten sub-problems (one for each go). The model objective is to minimize the pilot tardiness for all recurrent training requirements. The model developed by Newlon creates two feasible schedules in a matter of minutes, but as noted above he assumes an unlimited supply of reserve pilots. Furthermore, the three pilot qualifications he uses could be split further, as flight leads are generally classified as either 4-ship Flight Lead (4-FL) or 2-ship Flight Lead (2-FL) [16].

Yavuz' research also studies on a TUAF squadron, but takes a readily available weekly schedule and only performs pilot assignment [30]. However, he has taken note of some of Newlon's recommendations and accounts in his research for the difference between 4-FL and 2-FL. On top of that, he adds to each pilot a qualification which states in which weather conditions they are allowed to fly. The weather forecasts are not highly accurate, since they have to be forecasted a week in advance. Yavuz specifies that all pilots should maintain their currencies and perform some yearly checkflights. He does not take into account any sortie abort or rescheduling factors or dynamic resource availability. Yavuz splits his problem into four phases and the model objective is to maximize a scoring function which is based on pilot time to currency expiry and balancing sorties over the pilots. In the first phase of his model, the scheduler manually inputs high priority missions. In the second and third phase, a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic (3.2.2) is used to assign instructor pilots and other pilots respectively. During the last phase, the scheduler evaluates and manually adjusts the schedule if necessary. The main benefits of the developed method over fully manual scheduling are that the total amount of training events is balanced better between the pilots and that feasible assignments are found faster. The downside to this work is that it is an assignment problem rather than an integrated scheduling and assignment solution [30].

The goal of the research done by Yavuz was to construct an optimal daily schedule and assign pilots to the schedule for recurrent training missions in a TUAF squadron [4]. The planning environment and constraints are in many ways the same as that of Newlon and Yavuz. Additionaly, Erdemir adds the option that some of the aircraft are two-seated aircraft and that the backseats of those aircraft can only be manned by IPs. To solve the problem, Erdemir uses a GRASP heuristic (3.2.2). The main objective is to keep all pilots within their currency, while also maximizing sorties and aircraft usage. The model searches through all possible formations in a candidate list, picks the best formation, then updates the candidate list and continues. On top of that, the scheduler has the ability to manually schedule or exclude a specific pilot, formation or mission. Every iteration the model chooses another formation as the first best candidates, until all feasible schedules have been constructed. Afterwards all schedules are scored in accordance with the objectives and the best schedule is chosen. For small instances, the model is able to find optimal solutions within seconds. When Erdemir uses realistically sized data samples using twenty pilots however, the runtime increases to more than one hour. As such, Erdermir suggests to further investigate heuristics to reduce runtime. If reduced runtime can be realized, he also suggests to expand the scheduling horizon beyond his scope of a single day [4].

Due to the complex nature of the fighter pilot scheduling and assignment problem, all of the literature in this subsection so far considers building daily or weekly schedules. From an organizational point of view however, it would be beneficial to expand this planning horizon. On top of that, training requirements are often specified for a period of one year, so longer term planning is desired. Taks researches a case in the RNLAF where fighter pilots in an operational squadron have to complete a yearly recurrent training program [26]. Furthermore, the studied squadron also has to facilitate initial training to newly arriving pilots at the squadron. Taks aims to provide squadron schedulers with a baseline reference schedule, which would ease the schedulers' decision making process. Taks schedules and assigns all recurrent and initial training for every pilot for a whole year in three different ways. In each of his approaches he splits his problem into multiple sub-problems which are solved individually, either manually or with a MIP. It is impressive that he succeeds to find a yearly schedule within several minutes, but a lot of simplifications have been made to reduce the complexity of the problem. On top of that, when the required number of sorties increases the model is no longer able to find feasible solutions. In order to make Taks' work come closer to reality again, extensions should be made with respect to, for instance, dynamic resource availability. Despite Taks' effort, his model

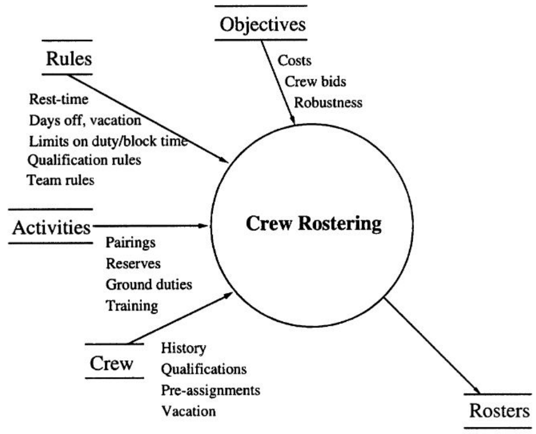


Figure 2.2: Representation of the airline crew rostering problem [13].

has not been adapted by the RNLAF [26].

Guljé studies almost the same problem as Taks, but adds more complexity to the problem and runs the yearly planning process as a simulation [9]. Apart from scheduling and assigning the flight sorties, he also plans simulator duties and he adds the requirement of night-flying into the recurrent training. On top of that, he makes the environment more dynamic by accounting for influences by weather, sickness, leave and other probabilistic factors. The objective of his model is to have as many pilots as possible achieve full completion of the yearly training program, by scheduling one month at a time. To create the schedules. Guljé uses a MIP which is based on the Generalized Assignment Problem (GAP) and uses priority scores based on pilot currencies and mission complexity. On the other hand of the spectrum than Taks, Guljé concludes that his model might be too complex. Computation times are in excess of multiple hours and completion of the training program is under par. He believes further research could be dedicated towards simplifying the model while at the same time looking into aircraft availability. In spite of his conclusion, Guljé still feels like his method holds a lot of potential, but like Taks, Guljé did not see his model adapted by the RNLAF [9]:

> "... the performance of the method is under par, we believe that there are still many aspects which can be further investigated, and which may possibly improve performance. Thus the method could be seen as a first step for creating monthly schedules in this largely restricted scheduling problem." [9]

2.2. Scheduling training in civil aviation

It goes without saying that pilots in civil aviation also have to meet requirements in able to maintain their level of training and that the environment in which this training has to be scheduled is complex. A visualization of the complexity of such a problem is given from Kohl and Karisch in Figure 2.2. The differences between fighter pilot training and civil pilot training however, are substantial. The main difference is that the training curriculum for civil pilots mainly consists of classroom and simulator sessions in between their regular flight schedule, while the training of fighter pilots essentially is their flight schedule, with some extra simulator sessions in between. As a result, cost induced by training in civil aviation not only comes from assigning resources, but also from the fact that each day a pilot is in training, the pilot is not available to part-take in the regular flight schedule. Yu et al. state that "the cost of training is measured primarily by the amount of time pilots spend in training and secondarily by the favourable assignment of training resources." [31] Differences between fighter pilot training scheduling and civil pilot training scheduling exist in many more aspects, but it is the similarities that we are interested in, not the differences. This section provides a brief overview of literature that has studied the scheduling or assignment problem (or a combination of both) of aircrew training within civil airlines and which has aspects that resemble the military environment.

Qi et al. investigate a class scheduling problem in the specific case of Continental Airlines [20]. Twice a year at Continental Airlines, a bidding process takes place, the result of which are numeral transitions of pilots between aircraft fleet, qualifications and bases. Consequently, many of those transitioning pilots have to attend training sessions in order to be qualified for their new position. The research of Qi et al. is concerned with scheduling those classes and assigning the pilots to the classes, where each schedule spans roughly one month. Each class has a predetermined number of candidates and a template that it has to follow. To minimize the cost induced by pilots' unavailability for the flight schedule, the goal of Qi et al. is to minimize the total weighted footprint of the classes. To solve the optimization problem, they created a Branch-and-Bound (B&B) algorithm (3.2.1) which is assisted by a rolling horizon approach (3.1.4). Qi et al. state that their model is able to solve the problem and obtain high quality schedules within several minutes, but it has to be noted that crew availability and detailed crew assignment is not taken into account [20].

Also within Continental Airlines Yu et al. take the research done by Qi et al. a step further and aim to find an integrated approach to allocating crew and training resources within Continental Airlines in an effective manner [31]. The process that Yu et al. aim to optimize is also that of the result of the bidding process. They divide the problem into four modules, the last of which is the training optimization module, which solves an aircrew training scheduling problem. This module takes as input training curricula, existing schedules and device and instructor availability and has as output training schedules, instructor schedules and device schedules (Figure 2.3). In addition to Qi et al. and nearly all the research in Section 2.1, Yu et al. also take crew availability due to vacation in account, which is an important improvement. They use the B&B (3.2.1) and rolling horizon approach (3.1.4) developed by Qi et al. to develop a class schedule and additionally have developed a MIP to perform detailed crew assignments. Yu et al. state that using their solver, the process which used to take weeks is now solvable within hours with huge savings in costs and more effective allocation of training resources [31].

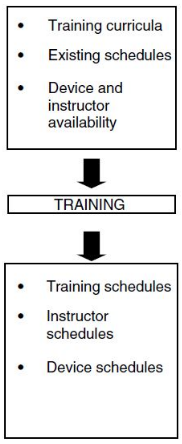


Figure 2.3: The training optimization module from Yu et al. [31].

Holm then takes the models proposed by Qi et al. and Yu et al. and adjusts them to fit her research problem within legacy carrier Scandinavian Airlines [11]. She follows the recommendation of Yu et al. to include recurrent training in the problem: she looks to schedule transition and recurrent training with the objective of minimizing costs. One of the main challenges in Holm's research comes from the fact that she wants to obtain a schedule for a whole year. According to Holm, cost is induced by the way pilots transition, training itself, pay protection systems and shortages in the flight schedule. The problem is to schedule a known number of classes with a known number of attendants in fixed crew compositions. Holm designs the problem as a set-partitioned (3.1.3) MIP. The model proposed by Holm solves the problem in about 18 hours, resulting in savings in cost up to 10%. As however, training resource availability is not constrained and as crew freetime can be bought back to resolve shortages, applicability of her research loses ground quickly [11].

The research done by Van Kempen also researches yearly recurrent and transition training in a European legacy carrier (not mentioned by name) [29]. He bases the training demand on historical data which is provided by the carrier and isolates the problem for each aircraft type in the fleet. All training events Van Kempen schedules are defined by yardsticks, which state per event their simulator and instructor demand. One simplification he makes is that the length and composition of these yardsticks are not based on pilot rank, experience or position. Aside from the scheduling and assignment of those yardsticks, Van Kempen also dedicates part of his research to developing a model which is robust with regards to schedule disruptions. The main objective of the thesis is to minimize the costs. To solve the problem, Van Kempen combines a selection heuristic and a priority heuristic before entering a MIP. The objective of the MIP is to minimize a sum of costs induced by simulator, trainee and instructor costs. The model is able to find optimal solutions within five minutes. Van Kempen however notes that not all of these solutions are feasible, because the heuristics can select solutions which are feasible in theory but not in practice [29].

Kozanidis moves away from the transition training scheduling and focuses solely on the monthly assignment of recurrent training [14]. The goal of Kozanidis' research is to have pilots fulfill their recurrent training before their currency expires. Kozanidis assumes that a set of training sessions is already scheduled and therefore focuses on assigning aircrew to these training sessions. He aims to find optimal assignment to a monthly training schedule, based on crew preference and seniority relations and to minimize the number of unassigned crew. The research differentiates between planning simulator sessions, which need to be carried out by a Captain (CPT) and a First officer (FO), and classroom sessions, which can be attended by all sorts of crew compositions. He solves both problem with a set-partitioned (3.1.3) MIP that uses crew preference and seniority related scoring functions. The model he proposes performs well, which is backed by the fact that it is actually integrated into the AIMS Airline Software suite. Kozanidis claims that the problem he encounters is found in military aviation as well and that his model could be adjusted for military use [14].

2.3. Synthesis on the research context

Having outlined some existing literature in both military and civil aviation, we can now move on to find which areas of scheduling are still partially unexplored and how the problem briefly described in the introduction fits into the existing literature. Table 2.1 gives an overview of the characteristics of all of the literature discussed in Sections 2.1 and 2.2. The first two columns in Table 2.1 indicate whether the indicated research is either a scheduling problem, an assignment problem or a combination of both. The third column, 'type of training', states which type of training the research tries to schedule for. The column 'length of schedule' denotes the length for which a fixed schedule is made, which is different from the 'length of program' column. In the latter column, the total length of a training program is given. For example, Nguyen constructs weekly schedules with the long-term goal of completing a training program within 120 days. The second-to-last column indicates if the training to be scheduled has a fixed, precedented sequential syllabus or not. Finally, the last column represents the different pilot classifications which have to be taken into account during scheduling.

Table 2.1: Overview of the characteristics of the discussed literature.

	Scheduling	Assignment	Type of training	Length of schedule	Length of program	Sequential syllabus	Classifications
Military:							
Nguyen[17]	Yes	Yes	initial	1 week	120 days	Yes	Student Pilot (SP), IP
Aslan[1]	Yes	Yes	initial	1 day	25 weeks	Yes	SP, IP, Bandit
Newlon[16]	Yes	Yes	recurrent	1 week	90 days	No	IP, FL, WM
Yavuz[30]	No	Yes	recurrent	1 week	1 year	No	IP, 4-FL, 2-FL, WM
Erdemir[4]	Limited	Yes	recurrent	1 day	n/a	No	IP, 4-FL, 2-FL, WM
Taks[26]	Yes	Yes	recurrent & initial	1 year	1 year	No	IP, 4-FL, 2-FL, WM, SP
Gulje[9]	Yes	Yes	recurrent	1 month	1 year	No	4-FL, 2-FL, WM
Civil:							
Qi et al. [20].	Yes	No	transition	± 1 month	1 year	Yes	CPT, FO
Yu et al. [31]	Yes	Yes	transition	± 1 month	1 year	Yes	CPT, FO
Holm[11]	Yes	Yes	transition & recurrent	1 year	1 year	Yes	CPT, FO
Kempen[29]	Yes	Yes	transition & recurrent	1 year	1 year	yes	CPT, FO
Kozanidis[14]	No	Yes	recurrent	1 month	1 year	Yes	CPT, FO

The length of schedules at each iteration in Qi et al. and Yu et al. is based on the number of classes that are being scheduled and is not fixed in time.

It can be seen from Table 2.1 that almost all literature discussed, focuses on both the scheduling and assignment problems within the aircrew training scheduling problem. This makes sense in the way that every pilot has different availability and different training requirements, so only scheduling the training events in an optimal way would not always assure feasibility in the assignment problem and vice versa.

We have split up the research into research into military aviation and research into civil aviation. From the table, it becomes clear that both fields have their main focus on different types of training, which is a direct result from how these two different sectors operate. Fighter pilots are trained for a large range of different operations and therefore require a lot of recurrent training. Civil airliner pilots are only required to be able to safely operate their aircraft on standard flights, thus they require less recurrent training. Where fighter pilots normally keep operating the same aircraft type for a long time and thus have little to no need for type transition training, airline pilots want to move up in their positions in the fleet and therefore require more transition training. Consequently, transition training is the main focus of civil airline crew training problems. What all of the research in the military field however fails to mention, is that some transition training can be identified within fighter pilot squadrons when pilots move up in the ranks as well. For example, WM might be selected to upgrade to 2-FL, 2-FL to 4-FL and so on.

The literature is further characterized by the nature of the syllabus that has to be scheduled and the different classifications pilots can have. These two aspects, as well as the type of training to be scheduled are closely interrelated. In civil aviation the differentiation is made between CPT and FO. These captains and first officers all have to attend type conversion training, which follows a fixed sequential syllabus. The fact that there are only two types of pilots and that the syllabus is fixed, vastly reduces problem size and solution space when comparing to the problem of recurrent training scheduling for fighter pilots. Fighter pilots are generally classified as IP, 4-FL, 2-FL, WM or SP. All these pilots have to fulfill a recurrent syllabus which contains more training events than the transition training for airline pilots. To allow for more flexibility in scheduling these events, while preference exists, no specific order is defined for these training events. Therefore, problem size and solution space for scheduling recurrent training in a yearly training program for fighter pilots (like Taks and Guljé) increase quickly.

In terms of the characteristics as defined by Table 2.1, our research focuses on a scheduling and assignment problem for a yearly initial, recurrent and transition training program for IP, 4-FL, 2-FL, WM and SP, which is not strictly sequentially fixed. This closely resembles the research of Taks and Guljé, as their research as well as ours is based on fighter pilots in the RNLAF. While the work by Guljé and Taks is closest to our problem, overlaps exist with all the other literature presented. Therefore, Chapter 3 further explores the methodologies the different works above used to model and solve the problems at hand.

3

Methodologies for formulating and solving scheduling problems

Having identified previous research into scheduling and assignment of training in aviation, it is now important to consider which solution methodologies are often adapted in order to solve such problems. Therefore, in this chapter we discuss solution methodologies that have been used by the researchers mentioned in Chapter 2 as well as some additional methodologies. These additional methodologies are considered as well, because they have been used effectively in other fields of scheduling problems or because they seem to offer a good framework to build our FPTSAP on. Section 3.1 briefly explains the concept of MIP, before elaborating on multiple approaches that can be used to ease the solution process of a MIP problem. Subsequently, in Section 3.2, we elaborate on methodologies that can be used in order to solve a MIP model. Lastly, in Section 3.3, we conclude the chapter with a summary of our findings.

3.1. Mixed Integer Programming

In operations optimization and scheduling problems, the fundamental methodology to formulate most problems is MIP. In MIP an objective function is defined, which is in turn maximized (or minimized) while respecting a set of constraints and predefined parameters. An example of a generalized MIP is the GAP which is elaborated on in Subsection 3.1.1. MIP is a straightforward tool and the advantage is that the concept is relatively simple and can be widely applied. Applying MIP to real size real world problems however, often results in large problems with rapidly increasing computation times. In fact, crew scheduling is known as a NP-hard problem, which means that no solution method is known to solve the problem within polynomial time[2]. As such, crew scheduling problems and their MIP models are often reformulated in one of the ways described in Subsections 3.1.1-3.1.5 in order to make them manageable to the solution methodologies discussed in Section 3.2.

3.1.1. Generalized assignment problem

The generalised assignment problem in its most basic form is a very straightforward scheduling problem, in which a set of tasks has to be performed by a set of agents. Each agent-task combination results in a cost or profit corresponding to that specific combination. The most basic form of the generalized assignment problem is formulated as a MIP problem as

$$minimize \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{3.1}$$

$$subject\ to \sum_{j \in J} r_{ij} xij \leq b_i \qquad \forall i \in I \tag{3.2}$$

$$\sum_{i \in I} x_i = 1 \qquad \forall j \in J \tag{3.3}$$

$$x_{ij} \in \mathbb{B} \qquad \forall i \in I, j \in J \tag{3.4}$$

for I the set of agents, J the set of tasks, c_{ij} the cost associated with agent $i \in I$ performing task $j \in J$, r_{ij} the resources agent $i \in I$ requires to perform task $j \in J$ and b_i the amount of resources available to agent $i \in I$

[23]. The idea of applying the general assignment problem to a scheduling problem or workforce planning problem is widely accepted, as Öncan details in his overview of applications of the general assignment problem in scheduling and other areas [32].

Both Yavuz and Guljé ([30], [9]) link fighter pilot scheduling to the GAP, but Guljé is the only one to actually base his MIP on it. Instead of minimizing cost, he seeks to maximize value. He defines the value as a priority score to determine at every scheduling decision which is the most important mission to schedule. This priority score is a function of how many times a pilot should execute a mission, the current currency of the pilot and how many times he had executed the mission in the previous cycle. This scoring function provides a relatively simple yet clear categorisation of which missions are most important and could also be applied to our problem. He further expands the GAP with more constraints with respect to go restrictions, sequencing restrictions, formation restrictions and more.

In the civil domain, all of Holm, Kozanidis, Van Kempen, Qi et al. and Yu et al. use a form of the GAP ([11], [14], [29], [20], [31]). However, all their cost or value parameters are based on either the cost of pilot absence in the flight schedule or the combined cost of instructor or simulator capacity usage. Given the nature of our fighter pilot problem where the cost of each sortie is the same and influence on the flight schedule is not a factor, these parameters are not of our interest.

3.1.2. Creation of sub-problems

Dividing a problem into several sub-problems is a simple method to improve computational efficiency within MIP, but it also has a downside. When one divides the problem into sub-problems, one should consider how this affects the performance of the model. The creation of sub-problems might have as a consequence that an optimal solution can no longer be found. One way to split a main problem into sub-problems is by defining the sub-problems in such a way that they can be solved simultaneously, as has been done for the airline recovery problem by Petersen et al.[18] A simpler method is to split the main problem into several sub-problems which can then be solved sequentially. This methodology has been used by Aslan, Erdemir, Taks and Yu et al.

Aslan divides his problem into three problems according to the pilot positions. He solves the problem first for students, then assigns instructors to the students and lastly adds an additional pilot if this is necessary. This works efficiently for his problem, where he has to match a single student and an instructor. If the problem would capture more complex formations and pilot qualifications however, the later sub-problems might become infeasible[1].

Erdemir divides his problem based on the most crucial tasks. In the first stages, he enforces critical tasks to be carried out by certain pilots. However, this might come at the cost of sacrificing optimality in the later stage, because certain pilots are no longer available[4].

Taks chooses to split his problem in time. During each sub-problem only missions in a specific category can be flown. He makes this decision based on practical implications and for the sake of continuity in the training program. As a result, the schedules his model provide are clear, but not by definition optimal[26].

Yu et al. make the logical decision to split their model up into two sub-problems based on model functionality. The first sub-problem schedules the classes and simulator sessions, while the second sub-problem does the detailed assignment of the crew members. In their approach they claim this two-stage approach comes at a cost of sacrificing minimal solution quality. The challenge in this approach is to construct a schedule which allows for feasibility in the assignment phase[31].

3.1.3. Set-partitioning

Another way to simplify a problem, much like dividing the problem into sub-problems, is to apply set partitioning to the problem. Set partitioning is defined by Garfinkel and Nemhouser as:

> "...considering a set of S of m elements to be partitioned into subsets chosen from a prescribed family $\{S_1,...,S_n\}$ of subsets of S. The subset S_j, $j = 1,...,n$ is represented by the binary vector A_j where $a_{ij} = 1$ if $i \in S_j$ and 0 if $i \notin S_j$."[6]

Effective set partitioning can reduce the problem size and solution space significantly even in the preprocessing phase and can often be applied by just using common sense. For example, in an aircrew training scheduling and assignment problem one could apply set partitioning to the pilots by dividing them into different sets in accordance with their qualifications as has been done in almost all the research discussed in

Chapter 2. Subsequently, if only pilots with certain qualifications are allowed to fulfill a specific duty or mission, the model only has to consider this set of pilots to assign to that mission and the number of variables can be decreased. Widely applied (sub)sets in fighter pilot scheduling are to divide missions into sets according to their formation size or duration or to divide pilots into sets according to which training programs and currencies they have to fulfill. A detailed example of set-partitioning applied in the Vehicle and Crew Scheduling Problem (VCSP) is given by Mesquita and Paias[15]. They define the decision variables in the problem as binaries instead of integers. This, combined with their definition of tasks and the fact that they partition the set of trips into multiple subsets of so-called deadhead trips, results in a decrease in the number of constraints.

3.1.4. Rolling horizon

The rolling horizon approach could be seen as a specific case of solving a model in several sub-problems. Instead of trying to find a solution for the whole problem at once, when a rolling horizon is used in modelling, the whole scheduling horizon which should be considered within a problem is split into discrete time periods. At each iteration, the problem is solved for the current time period but it can take into account information from past time periods or forecast data from upcoming time periods, often known as the prediction horizon. This process is schematically visualized by Silvente et al. in Figure 3.1 [25]. The challenge with this approach lies within determining how long the control and prediction horizons should be and in how to obtain forecast data from the prediction horizon. Rolling horizon approaches have been used by Qi et al. and Yu et al. and Guljé.

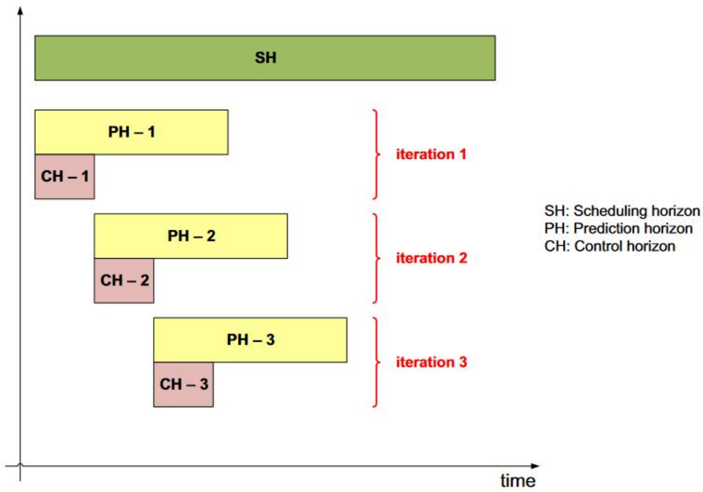


Figure 3.1: Schematic visualization of the concepts associated with rolling horizon approach [25].

Qi et al. and Yu et al. use the same implementation of the rolling horizon approach. Their objective is to schedule n classes, indexed in non-decreasing order of their earliest start dates, over a total scheduling horizon T. They argue that two classes that do not have any overlap, can only impact each other through a third class. Considering class 1, the further you would go down the list, the less likely this class is to have any overlap with the next classes. Therefore, at each iteration they only consider scheduling h classes, fix the optimal schedule of the current class given the h classes to be scheduled and then move on to the next iteration. They conclude that if they want to construct yearly schedules within a few minutes, while preserving solution quality, h should equal 4 or 5 ([20], [31]).

Guljé also uses a rolling horizon approach, but does so in an adjusted manner. At the beginning of each month m, Guljé already has a schedule which he developed at the beginning of month $m-1$. Guljé forecasts how factors like weather or pilot absence will influence the schedule completion in the upcoming month m (prediction horizon) by simulation. He then uses the results of this simulation to schedule for the subsequent month $m+1$ (control horizon). Guljé does not report on the performance of his rolling horizon implementation, as he has not benchmarked it against an alternative [9].

3.1.5. Benders decomposition

Benders Decomposition (BD) is a decomposition method that is often used in scheduling and assignment, as can be seen from the literature review by Rahmaniani et al. [21]. In BD the problem is split into a master problem and a subproblem, which are solved in turns iteratively. The master problem, which is initially unconstrained, is solved independently from the subproblem and the results are input as fixed variables into the subproblem. With these variables fixed, now the subproblem is solved. If it is unfeasible, an infeasibility cut is added to the master problem. If the sub-problem finds an optimal solution, an optimality cut is added to the master problem. With this new information, the master problem enters the next iteration. A detailed explanation of the Benders decompostion algorithm is given by Taskin in his chapter 'Benders decomposition' in the 'Wiley Encyclopedia of operations research and management science' [27].

While none of the research mentioned in Chapter 2 uses BD to solve their problem, it is a useful approach for large-scale problems that contain so-called complicating variables. When fixing these variables, the problem that remains becomes significantly easier to solve[21]. Considering this, Benders decomposition could be of use for a scheduling and assignment problem. One could see creating and fixing a schedule as fixing the complicating variables and with a fixed schedule, the assignment problem becomes significantly easier.

3.2. Solution methodologies

Now that some approaches have been discussed which can potentially speed up the solution process, we can elaborate on some of the methods that can be used to actually compute the solutions. One of the most popular methodologies is the B&B algorithm which is discussed in Subsection 3.2.1. Apart from the B&B algorithm, several heuristic methods are discussed in Subsection 3.2.2.

3.2.1. Branch and Bound

Clausen states that "B&B is by far the most widely used tool for solving large scale NP-hard combinatorial optimization problems." [2] A B&B algorithm is able to search the whole solution space of a problem. This however is also its greatest pitfall. If one does not carefully specify the upper and lower bound and fathoming criteria at each iteration, the danger arises that a B&B algorithm turns into an exhaustive search of the whole solution space. The idea behind B&B is best described by Clausen in the paragraph below and is visualized in Figure 3.2. This process continues until no unexplored solution space remains and at that point the best solution is offered as the optimal solution. For a more detailed description and examples, one can consult the work of Clausen[2]. Considering the previously discussed literature, it becomes apparent that B&B is indeed a popular tool; it is applied in the research of Holm, van Kempen, Qi et al. and Yu et al.

> "At any point during the solution process, the status of the solution with respect to the search of the solution space is described by a pool of yet unexplored subsets of this and the best solution found so far. Initially only one subset exists,namely the complete solution space, and the best solution found so far is ∞. The unexplored subspaces are represented as nodes in a dynamically generated search tree, which initially only contains the root, and each iteration of a classical B&B algorithm processes one such node. The iteration has three main components: selection of the node to process, bound calculation, and branching."[2]

Qi et al. develop their own B&B algorithm. At each node D the current calendar day t and the partial schedule PS are considered. At each node, every class that has not yet finished is said to be active. One can move from note D to successor node D' by attempting to schedule an event, which is either a day off or a training event. To fathom nodes that are either infeasible or sub-optimal, node-elimination constraints and a lower-bound calculation are used. The node-elimination constraints are based on rules such as that a training event can not take place during weekends or that a day off results in violating the latest start date of a class. Lower-bound calculation is done by estimating the number of future days off in each partial schedule. Qi et al. state that they had first tried to solve the problem using CPLEX. Due to the nature and size of the problem, they were unable to find feasible solutions. With their own B&B algorithm however in combination with the rolling horizon approach (Subsection 3.1.4), they were able to find feasible yearly schedules within a few minutes [20].

Contrary to Qi et al. and Yu et al., all of Guljé, Holm and Taks solve their problem with a readily available commercial solver. It is important to note this aspect at this point, because almost all commercial solvers use a sophisticated B&B algorithm to solve the MIP models they are confronted with.

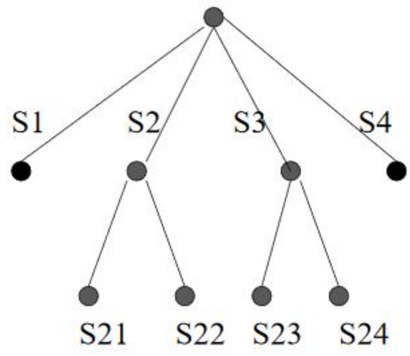


Figure 3.2: An example of a B&B search tree [2]. Note that subsets $S23$ and $S24$ should actually be indexed $S31$ and $S32$.

3.2.2. Heuristics

Even though commercial solver use highly complex and efficient B&B algorithms, solving large-scale problems to an exact and optimal solution often still results in excessive runtimes. To overcome this problem, we now take a look at heuristics. A heuristic is described by Salcedo-Sanz as follows.

> "Different from exact methods, which guarantee an optimum solution for the optimization problem, heuristic methods try to obtain a good (though not necessarily the optimum) solution. In other words, a heuristic algorithm is a problem-solving method which tries to obtain good-enough solutions for a given optimization problem at a reasonable computational cost, but without guaranteeing either their feasibility nor optimality. the majority of heuristics approaches for optimization are based on specific characteristics of the problem, and in a local or global search carried out with some specific methods."[24]

In other words, heuristics are just smart tricks that attempt to decrease the computation time in which an acceptable solution to very large problems can be found. To illustrate the effect of heuristics, we can consider the work by Kim and Kim. Their model needs 3 hours to solve using exact optimization methods, but solves within seconds when applying their heuristic [12].

GRASP The GRASP is an iterative process which is divided into two phases: the construction phase and the local search phase [22]. In each iteration, the construction phase picks a random feasible candidate solution and adds this to the partial solution. The incremental cost from adding this candidate to the solution is then evaluated, so that a Restricted Candidate List (RCL) is formed. This RCL consists of the best candidates. A random candidate from the RCL is then fed into the partial schedule and the lists are updated until a feasible schedule is found. Pseudo code for the construction phase can be found in Figure 3.3. During the next phase, the local search phase, the neighborhood of the schedule is investigated until a local optimum is found. Pseudo code for this phase is given in Figure 3.4. As with every local search procedure, the effectiveness of GRASP is determined by the definition of the neighborhood and the neighborhood search algorithm. The neighborhood search may either be based on the first-improving or best-improving strategy, but Resendo and Ribeiro state that both strategies often lead to the same final solution, while the first-improving strategy results in better computation time [22]. GRASP has already been used in the fighter pilot scheduling and assignment problem by Yavuz and by Erdemir.

Yavuz implements a simplified adaptation to the GRASP heuristic, which does not include a local search phase. To assign pilots to missions, he uses GRASP to assign IPs and the remaining pilots separately, but both methods are the same. Before every assignment, every pilot receives a grade based on multiple factors which indicates how preferable it is to assign said pilot to the current mission. A number of these pilots is put into the RCL, based on a weighting factor α which is set by Yavuz. Then a random assignment is made from the RCL, the grades are updated and this continues until either no pilots are left or no missions are left to schedule [30].

Erdemir uses a slightly different GRASP implementation which consists of multiple iterations. Every time a scheduling decision has to be made, a candidate list is constructed which consists out of feasible pilot formations. All candidates receive a random scoring function and are sorted in decreasing order. A candidate

```
procedure Greedy_Randomized_Construction(Seed)
    Solution ← ∅;
    Evaluate the incremental costs of the candidate elements;
    while Solution is not a complete solution do
        Build the restricted candidate list (RCL);
        Select an element s from the RCL at random;
        Solution ← Solution ∪ {s};
        Reevaluate the incremental costs;
    end;
    return Solution;
end Greedy_Randomized_Construction.
```

Figure 3.3: Pseudo code for the GRASP construction phase [22].

```
procedure Local_Search(Solution)
    while Solution is not locally optimal do
        Find s' ∈ N(Solution) with f(s') < f(Solution);
        Solution ← s';
    end;
    return Solution;
end Local_Search.
```

Figure 3.4: Pseudo code for the GRASP local search phase [22].

formation is chosen from the list and the list is updated by removing formations that have become infeasible as a result of selecting the first candidate. This continues until no feasible formations remain or no resources remain. Then the algorithm will continue to the next iteration. At the first step during iteration n, the nth candidate is chosen from the ordered list. During all subsequent stepts, the first candidate is chosen. Once the algorithm has looped through all n formations, the schedule with the best overall score is fixed [4].

Tabu Search Tabu Search (TS) is a meta-heuristic first introduced in 1986 ([8]) that effectively is an extension of classical local search methods like GRASP [7]. During a tabu search, non-improving moves are allowed to move away from local optima. Complementary, to prevent the search method to cycle back to these local optima, tabus are implemented. The tabu-list keeps track of a list of disallowed moves[7]. Every time a certain mutation is made to the schedule, the exact reverse of this mutation can be added to the tabu-list. This new entry is then added to the top of the tabu-list and all other entries are pushed down and consequently the last entry is deleted[19]. The search can be terminated when either a specific threshold value has been met, after a predetermined number of iterations or after a predetermined number of iterations without improvement in the objective function. A simple and straightforward example of TS implementation is given by Pinedo ([19]), while a clear template of the method by Gendreau is seen in Figure 3.5. While we have not found any literature which implements TS in (fighter) pilot training scheduling & assignment problems, it could possibly be applied in a similar manner as Yavuz and Erdemir implement the GRASP heuristic, since TS is in essence an extension of GRASP.

Dispatching rules Dispatching rules find their origin in the field of Machine Scheduling (MS). Dispatching rules are a means to select the most important task to be carried out first. An example of a dispatching rule would be Least Flexible Job (LFJ), which stats that the job which is the least flexible to perform has to be scheduled first. In fighter pilot scheduling, this could for instance be the job that has to be flown by the largest amount of aircraft. Both Aslan and Nguyen use dispatching rules in the construction phase of their model. Aslan uses a scoring function which combines three dispatching rules to construct his schedule. The scoring is based on LFJ, Largest Number of Successors (LNS) and Largest Number of Modes (LNM) dispatching rules [1]. Nguyen also implements dispatching rules in a scoring function, but lets the scheduler decide between one of three rules: LFJ, Longest Processing Time (LPT) and MS[17]. For a more detailed description of Minimum Slack (MiS) in general and dispatching rules in specific, one can consult Pinedo's book 'Scheduling. Theory, algorithms and systems.' [19].

Notation

- $\boldsymbol{S}$, the current solution,
- $\boldsymbol{S^*}$, the best-known solution,
- $\boldsymbol{f^*}$, value of S^*,
- $\boldsymbol{N(S)}$, the neighborhood of S,
- $\boldsymbol{\tilde{N}(S)}$, the "admissible" subset of $N(S)$ (i.e., non-tabu or allowed by aspiration).

Initialization

Choose (construct) an initial solution S_0.
Set $S := S_0, f^* := f(S_0), S^* := S_0, T := \emptyset$.

Search

While *termination criterion not satisfied* do

- Select S in argmin $[f(S')]$; $S' \varepsilon \tilde{N}(S)$
- if $f(S) < f^*$, then set $f^* := f(S), S^* := S$;
- record tabu for the current move in T (delete oldest entry if necessary);

endwhile.

Figure 3.5: Template of the TS heuristic method [7].

3.3. Conclusion on methodologies

Almost all recent literature that discuss pilot training scheduling & assignment, use MIP to formulate their problem. A straightforward MIP model however will often result in computation times of multiple hours, if feasible solutions can be found at all. Therefore, some additional formulation approaches can be applied to help structure a problem and the input data into efficient formats which could ease the computation time. For example, we could take GAP as a basic framework. The agents defined by the GAP would then correspond to pilots, the tasks to missions and the resources are available aircraft and training slots. Additional constraints would have to be developed to fully fit the problem, as done by Guljé [9]. Additionally, previous literature has proven that it is beneficial to split the problem into sub-problems or to apply set-partitioning or rolling horizon. Lastly it has been noted that the nature of the FPTSAP might also be suitable to approach with BD.

Subsequently, two widely applied solution methodologies to solve MIP models have been discussed. B&B is the go-to exact optimization solution methodology and is implemented byy almost all commercial solvers. If however, B&B falls short in terms of runtime, one could resort to heuristics. Heuristics could drastically speed up the solution process while sacrificing minimal solution quality. Both methodologies have been used to solve crew scheduling problems before and have proven their worth.

When problem size continues to grow or when allowed computation time for a model is very short, exact optimization through B&B algorithms solely might no longer be preferable. Therefore we also explored some heuristics. Heuristics methods rely on smart algorithms that can find high quality, but sub-optimal solutions to large-size problems within relatively short time. Numerous heuristics exist, but among the most widely used in scheduling problems are GRASP and the implementation of dispatching rules in a construction heuristic. At the same time, TS is also an option as a smarter alternative to GRASP.

In this chapter, basic modelling and formulation approaches and solution methodologies in previous research have been presented, as well as some formulation approaches and solution methodologies that could be applicable to the FPTSAP, but to the best of our knowledge have not been used before.

4

Conclusion & Contribution to research

Chapter 2 presented an overview of the literature that is available in the field of aircrew training scheduling and assignment. Aircrew training is a costly factor for airlines and improving training scheduling could be highly beneficial; Yu et al. report that their work spared Continental Airlines more then 10 million dollars in a year [31]. Nevertheless, literature on aircrew training scheduling and assignment is not as widely available as one would expect. Still, we managed to find several relevant research papers and master theses, all of which focus on their own specific problem in the research context. However, it can be concluded from Chapter 2 in general and from Table 2.1 in Section 2.3 in specific that none of the works measures up to all of the following criteria at the same time:

1. The research is focused on a military fighter squadron.
2. The research aims on developing a baseline schedule for a whole year.
3. The research aims on scheduling initial, recurrent and transition training.
4. The model can obtain near-optimal results for a realistic use case.
5. The model used can solve a realistic use case within minutes or a few hours.

For example, if we consider the work done by Holm, we can see that her work did focus on building a yearly schedule while remaining close to reality thereby satisfying requirements 2 and 4. However, her research is focused on transition and recurrent training within a civil airline and the model needs 18 hours to solve, so her research does not comply with criteria 1, 3 and 5 [11]. The works that come closest to fulfilling all requirements are that of Taks and Guljé. Taks, a fighter pilot himself, nonetheless noted in a personal conversation that his research contains too much simplifications and assumptions, therefore failing to fulfill requirement 4 [26]. Guljé's work on the other hand meets requirements 1 up till 3, but he himself notes in his work that the results are under par and that computational efficiency could be better [9]. We can therefore conclude that to the best of our knowledge, no literature exists that succeeds to fulfill all five of these requirements and as such we have identified a gap in the research which can be described by the following research objective:

> The research objective is to develop a realistic model that can obtain near-optimal yearly schedules for initial, recurrent and transition training in a military fighter squadron within several minutes to a few hours.

As to the best of our knowledge no other research has yet fulfilled this objective, we must also develop an adequate methodology to approach this problem. In Chapter 3 we discussed the methodologies that are most commonly used in scheduling problems. Initially we formulate the model as a MIP. As we have to deal with a total scheduling horizon of one year, the number of variables within the model will rise quickly, thus we have to come up with methods to work around this, which have been provided in Section 3.1. Subsequently, Section 3.2 has detailed methodologies to solve the MIP model. In a later stadium of our research it will become clear which (combination) of these is the most suitable.

As to focus the research and provide a clear path throughout the whole research and writing process, it is imperative to have a clear and concise research question. This research question follows from combining the research objective and the problem statement given to us by our client, the RNLAF. Since writing a master's thesis is a complex and lengthy process, it is wise to split the main research question into several sub-questions.

> How can we develop a decision support tool that is able to aid Royal Netherlands Air Force (RNLAF) fighter squadron schedulers in constructing a baseline yearly schedule?
>
> - What are the current challenges in scheduling training flights within a typical RNLAF fighter squadron?
> - What are the requirements of the initial, recurrent and transition training programs?
> - What resources are available and how is the planning environment constrained?
> - What model form is suitable to find a solution to this problem within the specified requirements?
> - What methodology is suitable to solve the model and balance runtime and solution quality?

In the end, the project can be concluded by clearly answering each sub-question and if by doing so, one can also provide an answer to the main research question in such a way that the research objective has been met and that the gap in the research has been closed.

III

Supporting work

5
FPTSAP model

The solution methodology for the FPTSAP model is discussed in detail in the scientific paper in Part I. This chapter serves as a guide to understand the flows and processes within the model. First, Section 5.1 gives some insight on the setup that is used to write the model, run the model and obtain the results. In Section 5.2 the model architecture is explained and put into context of the solution methodology. Section 5.3 provides an explanation of the pre-processors that are used in the model.

5.1. Model Setup

The model is written in the Python programming language, version 3.7. As such, for more information with regards to the terms and semantics used to describe the model, one can consult the Python documentation [5]. For the ease of programming and debugging, the Spyder IDE is used to write the model script [3]. To solve the optimization problems in the model, Gurobi is used [10]. Advertised as "the fastest solver in the world", Gurobi uses a combination of a complex B&B algorithm and several heuristics to solve the models it is offered. The Master Problem (MP) solves with all parameters set at their default value. For every Sub Problem (SPr) however, a runtime limit of 20 seconds is imposed. In general the SPrs solve to optimality within this time limit, but exceptions can occur where a SPr has difficulties to converge from a solution with an optimality gap of less than 5% to the optimal solution. To prevent the model from getting stuck within such a SPr, the time limit is set. Information on the machine used to run the model and obtain the results is found in Table 5.1.

Table 5.1: Specifications of the used system.

Operating System	Windows 10 Home
Processor	Intel Core i5-7200U 2.5GHz
RAM	8.00 GB
System Type	x64-based

5.2. Model Architecture

An overview of the high-level architecture of the FPTSAP model is given in Figure 5.1. The core of the model is marked yellow, required input is marked blue, functions that serve as pre-processors are pink, output is represented by green and the control sequence that handles the number of runs is maintained white.

The core of the model is defined as a Python class, that possesses multiple methods. Before the program is run, the user has to specify the values for the user-defined import parameters defined. Then, when the model is initiated, the Init method assigns these parameters to an attribute and also defines other attributes that are used later. The Read method reads the two workbooks that contain the input data on the scenario. Subsequently, all data is structured in such a way that it is easily accessible by the Prep method and the accompanying functions that act as pre-processors. The contents and functionalities of these pre-processors are elaborated upon in Section 5.3. Then, the Build MP method constructs the MP constructs the MIP that will

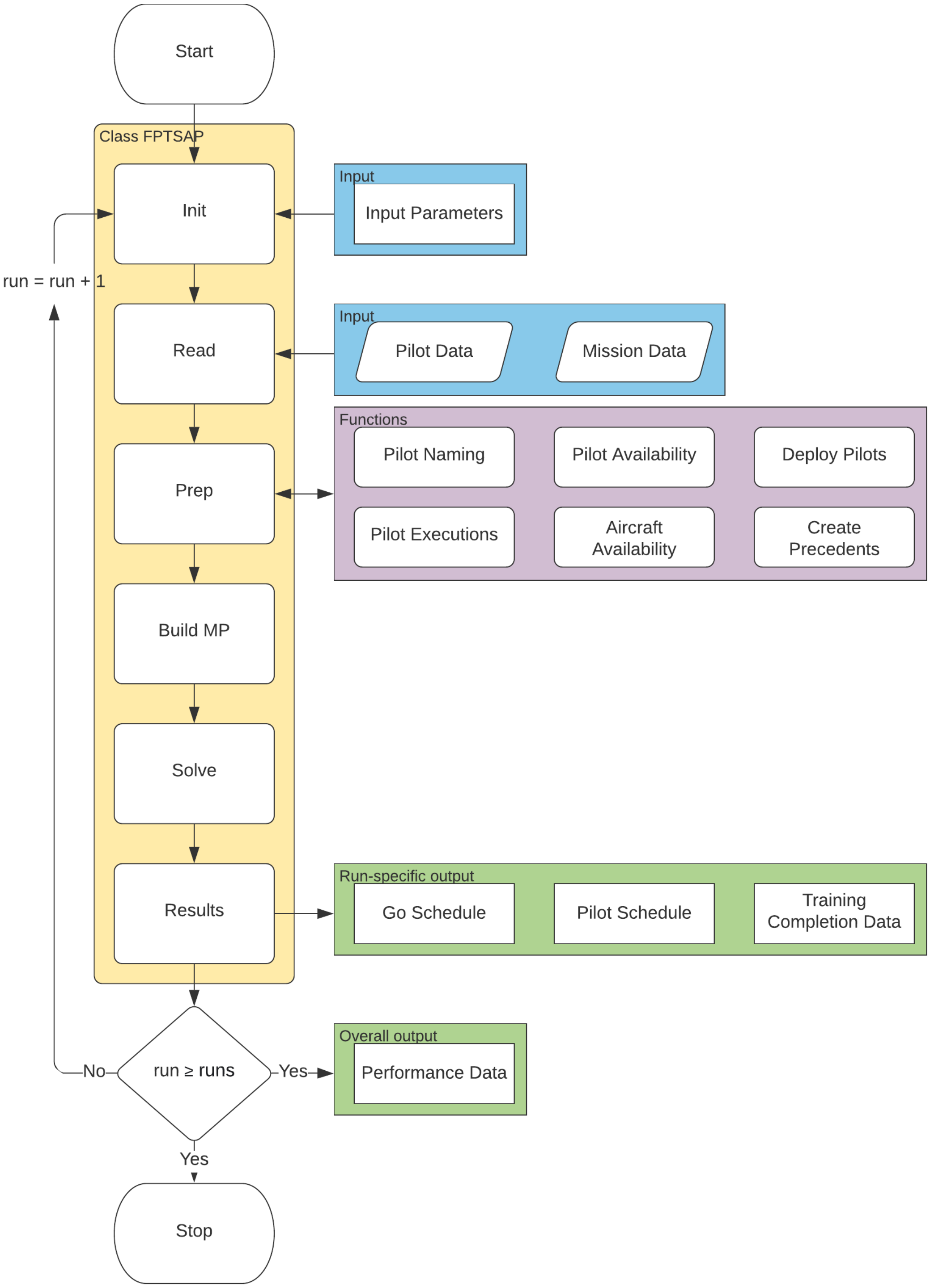


Figure 5.1: Flow chart of the FPTSAP model.

be passed on to Gurobi. Next, the Solve method takes care of the solution process, which is explained further in the next paragraph. The Results method processes the results for this iteration. Out of all iterations made, it decides which one offers the best solution. For this solution, the schedules and all relevant information on completed training are output. Afterwards, either a new run is started by returning to init and creating a new problem instance, or the full model is stopped and the combined data of all runs is output.

The Solve method is more complex as it also calls other methods. This process is visualized in Figure 5.2. It begins by calling the Gurobi solver to optimize the MP. Subsequently, the first SPr is constructed using the solution from the MP, which states what mission category is scheduled for the current week. The current SPr is then solved by Gurobi. Method SPr Results stores the results of the current SPr, such as the schedule and completed training per pilot. Additionally, it stores a MIP start for the initial conditions of this SPr. These MIP starts and the completed training data are used as input for future SPrs. When the results have been processed, the model checks if the current SPr was the last in the scheduling horizon. If this is not the case, the model moves to the next week and returns to the Build SPr method. If the scheduling horizon is finished, the results for the current iteration of the full scheduling horizon are stored by the method IP Results. This method also cuts off the current solution, so the MP should come up with another solution in future iterations. Subsequently, if the objective value for the current iteration is lower than the set limit, a new iteration is started by optimizing the MP again. If it was the last iteration, the model leaves the solve block and continues in Figure 5.1.

5.3. Pre-Processors

Pilot Naming This pre-processor assigns a unique identifier to each pilot, based on the total number of pilots, the qualification of the pilot and the number of pilots with that qualification. Normally, the model user could just input the actual names of the pilots. For this thesis however, the actual data from the RNLAF is classified and as such a semi-random scenario is used. Then, assigning an identifier like this is an easy manner to be able to distinguish the pilots.

Pilot Executions Taking as input the pilot data and mission data workbooks, this function calculates how many times every pilot should complete every specific training mission in order to obtain full completion of the assigned training syllabi. This information is obtained from the syllabi that are assigned to the pilot and the level of experience of the pilot.

Pilot Availability This function takes care of assigning days off to the pilots. If the FPTSAP model is used to build a schedule on a real case, this function would be obsolete, because the availability of the pilots would be known. In this case however, the function takes as input the pilots, the scheduling horizon and how many days off should be assigned to every pilot. For the reference scenario, every pilot is awarded one day off for every ten working days. This is roughly in line with the guidance set in the collective labor agreement for Dutch military personnel. These days off are randomly spread out over the scheduling horizon for every pilot.

Aircraft Availability Analog to the Pilot Availability pre-processor, this function maps the availability of the number of aircraft throughout the scheduling horizon, as no real (unclassified) data is available on the aircraft availability. In day-to-day practice however, it can be seen that it is common to have either 4, 6 or 8 aircraft available for training missions at every go. At the same time, the maintenance organisation aims to have a constant amount of aircraft available during a single week. Combining these aspects, this function takes the scheduling horizon in terms of weeks and makes either 4, 6 or 8 aircraft available, randomized over all weeks, but with an average of 6 aircraft. If however, there are deployments in the schedule, these weeks are automatically only assigned 4 available aircraft, since some aircraft are also used to carry out the deployment. These numbers are specific to the case, but could also be changed.

Deploy Pilots Depending on the size of the deployments, every deployment must be filled with a number of pilots of adequate qualifications. This pre-processor takes as input the size of the deployments and the number of deployment periods. Then for every deployment period, the function assures that enough pilots are scheduled to go on the deployment and that the pilots that are chosen have the right qualifications to meet the demands that are set for the deployment.

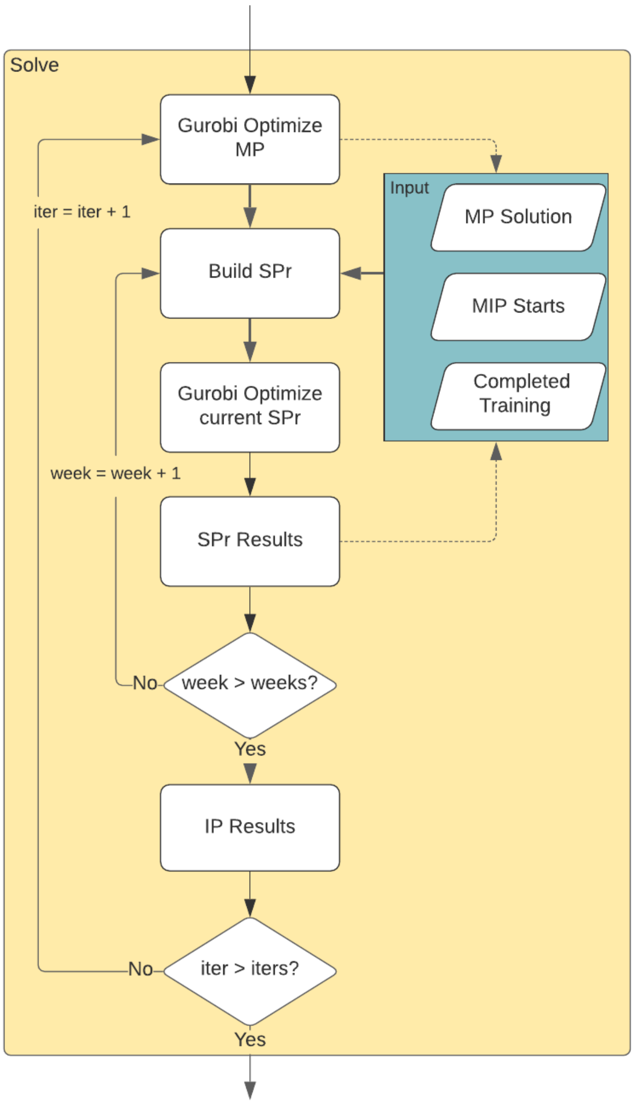


Figure 5.2: Detailed process of the Solve method from 5.1

Create Precedents In initial training as well as in transition training, certain missions have one or more precedents. This means that those missions can only be completed after having completed all of the required precedents. This function structures the missions and precedents in such a way that they can be easily read by the model, independent of how many precedents a mission has.

IV

Appendices

A

Scenario Data

This appendix provides an overview of the data that is used as input to test the FPTSAP model. Section A.1 gives an overview of the pilot data and the same is done in Section A.2 for the mission data.

A.1. Pilot Data

Tables A.1 - A.4 give an overview of the sets of pilots that are used in the different testing scenarios. Each pilot has a qualification, a status indicating his experience and is assigned one or more syllabi that he should complete. RT means a pilot has to complete recurrent training, IL means a pilot has to complete initial training, U2 means a pilot should complete transition training from WM to 2-FL and U4 means a pilot should complete transition training from 2-FL to 4-FL.

Table A.1: Pilot data for the scenario with 10 pilots.

P ID	Qualification	Status	Syllabi
1	IP	exp	RT
2	IP	exp	RT
3	IP	exp	RT
4	F4	exp	RT
5	F4	exp	RT
6	F2	exp	RT, U4
7	F2	exp	RT
8	WM	inexp	RT, U2
9	WM	inexp	RT
10	SP	inexp	IL

Table A.2: Pilot data for the reference scenario.

P ID	Qualification	Status	Syllabi
1	IP	exp	RT
2	IP	exp	RT
3	IP	exp	RT
4	IP	exp	RT
5	IP	exp	RT
6	IP	exp	RT
7	IP	exp	RT
8	F4	exp	RT
9	F4	exp	RT
10	F4	exp	RT
11	F4	exp	RT
12	F4	exp	RT
13	F2	exp	RT, U4
14	F2	exp	RT
15	F2	exp	RT
16	F2	inexp	RT
17	F2	inexp	RT
18	F2	inexp	RT
19	WM	inexp	RT, U2
20	WM	inexp	RT, U2
21	WM	inexp	RT
22	SP	inexp	IL
23	SP	inexp	IL

Table A.3: Pilot data for the scenario with 50 pilots.

P ID	Qualification	Status	Syllabi
1	IP	exp	RT
2	IP	exp	RT
3	IP	exp	RT
4	IP	exp	RT
5	IP	exp	RT
6	IP	exp	RT
7	IP	exp	RT
8	IP	exp	RT
9	IP	exp	RT
10	IP	exp	RT
11	IP	exp	RT
12	IP	exp	RT
13	IP	exp	RT
14	IP	exp	RT
15	F4	exp	RT
16	F4	exp	RT
17	F4	exp	RT
18	F4	exp	RT
19	F4	exp	RT
20	F4	exp	RT
21	F4	exp	RT
22	F4	exp	RT
23	F4	exp	RT
24	F4	exp	RT
25	F4	exp	RT
26	F2	exp	RT, U4
27	F2	exp	RT
28	F2	exp	RT
29	F2	inexp	RT
30	F2	inexp	RT
31	F2	inexp	RT
32	F2	exp	RT, U4
33	F2	exp	RT, U4
34	F2	exp	RT, U4
35	F2	inexp	RT
36	F2	inexp	RT
37	F2	inexp	RT
38	F2	inexp	RT
39	WM	inexp	RT, U2
40	WM	inexp	RT, U2
41	WM	inexp	RT
42	WM	inexp	RT, U2
43	WM	inexp	RT, U2
44	WM	inexp	RT
45	WM	inexp	RT, U2
46	SP	inexp	IL
47	SP	inexp	IL
48	SP	inexp	IL
49	SP	inexp	IL
50	SP	inexp	IL

Table A.4: Pilot data for the scenario with 75 pilots.

P ID	Qualification	Status	Syllabi
1	IP	exp	RT
2	IP	exp	RT
3	IP	exp	RT
4	IP	exp	RT
5	IP	exp	RT
6	IP	exp	RT
7	IP	exp	RT
8	IP	exp	RT
9	IP	exp	RT
10	IP	exp	RT
11	IP	exp	RT
12	IP	exp	RT
13	IP	exp	RT
14	IP	exp	RT
15	IP	exp	RT
16	IP	exp	RT
17	IP	exp	RT
18	IP	exp	RT
19	IP	exp	RT
20	IP	exp	RT
21	IP	exp	RT
22	F4	exp	RT
23	F4	exp	RT
24	F4	exp	RT
25	F4	exp	RT
26	F4	exp	RT
27	F4	exp	RT
28	F4	exp	RT
29	F4	exp	RT
30	F4	exp	RT
31	F4	exp	RT
32	F4	exp	RT
33	F4	exp	RT
34	F4	exp	RT
35	F4	exp	RT
36	F4	exp	RT
37	F4	exp	RT
38	F2	exp	RT, U4
39	F2	exp	RT, U4
40	F2	exp	RT, U4
41	F2	exp	RT, U4
42	F2	exp	RT, U4
43	F2	exp	RT
44	F2	exp	RT
45	F2	exp	RT
46	F2	exp	RT
47	F2	exp	RT
48	F2	exp	RT
49	F2	exp	RT
50	F2	exp	RT
51	F2	inexp	RT
52	F2	inexp	RT

Table A.4: Pilot data for the scenario with 75 pilots.

P ID	Qualification	Status	Syllabi
53	F2	inexp	RT
54	F2	inexp	RT
55	F2	inexp	RT
56	F2	inexp	RT
57	WM	inexp	RT, U2
58	WM	inexp	RT, U2
59	WM	inexp	RT, U2
60	WM	inexp	RT, U2
61	WM	inexp	RT, U2
62	WM	inexp	RT, U2
63	WM	inexp	RT
64	WM	inexp	RT
65	WM	inexp	RT
66	WM	inexp	RT
67	WM	inexp	RT
68	SP	inexp	IL
69	SP	inexp	IL
70	SP	inexp	IL
71	SP	inexp	IL
72	SP	inexp	IL
73	SP	inexp	IL
74	SP	inexp	IL
75	SP	inexp	IL

A.2. Mission Data

Table A.5 gives an overview of the set of missions that is used in the reference scenario. The most important characteristics of each mission are the amount of times it should be performed per syllabus, the mission category it belongs to, the mission size and the precedents to the mission. The abbreviations that denote the number of executions required per syllabus are: R1 for recurrent training by experienced pilots, R2 for recurrent training by inexperienced pilots, IL for initial training, DY for deployment work-up programs, U2 for transition training from WM to 2-FL and U4 for transition training from 2-FL to 4-FL.

Table A.5: Mission data for the reference scenario.

M ID	S	Blue size	Total size	Red mission	#R1	#R2	#IL	#DY	#U2	#U4	Prec	Same	MC	MC Alt
1	IL, RT	2	2	/	1	1	1	0	0	0	/	/	A1	/
2	IL, RT	2	2	/	1	1	1	0	0	0	1	/	A1	/
3	IL, RT	2	2	/	0	1	1	0	0	0	2	/	A1	/
4	IL, RT	2	3	35	1	1	1	0	0	0	3	/	A1	/
5	IL, RT	2	3	35	1	1	1	0	0	0	4	/	A1	/
6	IL, RT	2	4	35	3	3	1	0	0	0	5	/	A2	/
7	IL, RT	2	4	36	2	3	1	0	0	0	6	/	A3	/
8	IL	2	3	35	0	0	1	0	0	0	7	/	A3	/
9	IL, RT	2	4	36	1	2	1	0	0	0	6	/	A4	/
10	IL, RT	4	6	36	2	3	1	0	0	0	6	/	A2	/
11	IL, RT	4	6	36	2	2	1	0	0	0	10	/	A3	/
12	IL, RT	4	6	36	2	2	1	0	0	0	9	/	A4	/
13	IL	2	3	35	0	0	1	0	0	0	6	/	A2	/
14	IL	4	6	36	0	0	1	0	0	0	7	/	A3	/
15	IL	4	6	36	0	0	1	0	0	0	9	/	A4	/
16	IL	2	2	/	0	0	1	0	0	0	/	/	A5	/
17	IL	2	2	/	0	0	1	0	0	0	/	/	A5	/
18	IL, RT	2	2	/	1	2	1	0	0	0	/	/	G1	/
19	IL, RT	2	2	/	1	1	1	0	0	0	/	/	G1	/
20	IL, RT, DY	2	2	/	3	5	1	2	0	0	/	/	G2	/
21	IL, RT, DY	2	2	/	1	1	1	2	0	0	20	/	G2	/
22	IL	2	2	/	0	0	1	0	0	0	20	/	G2	/
23	IL	2	2	/	0	0	1	0	0	0	21	/	G2	/
24	IL	2	2	/	0	0	1	0	0	0	21	/	G2	/

Table A.5: Mission data for the reference scenario.

M ID	S	Blue size	Total size	Red mission	#R1	#R2	#IL	#DY	#U2	#U4	Prec	Same	MC	MC Alt
25	IL	2	2	/	0	0	1	0	0	0	22, 23, 24	/	G3	/
26	IL	2	2	/	0	0	1	0	0	0	25	/	G3	/
27	IL	4	4	/	0	0	1	0	0	0	26	/	G3	/
28	IL	2	2	/	0	0	1	0	0	0	22, 23, 24	/	G3	/
29	IL, RT, DY	2	4	36	3	4	1	2	0	0	28	/	G3	/
30	IL, RT, DY	4	6	36	3	3	1	1	0	0	29	/	G3	/
31	IL	2	2	/	0	0	1	0	0	0	22, 23, 24	/	G4	/
32	IL, RT	2	2	/	3	3	1	0	0	0	31	/	G4	/
33	IL	2	2	/	0	0	1	0	0	0	32	/	G4	/
34	RT	2	4	36	1	2	0		0	0	/	/	A4	/
35	ST	1	1	/	0	0	0	0	0	0	/	/	R	/
36	ST	2	2	/	0	0	0	0	0	0	/	/	R	/
37	ST	4	4	/	0	0	0	0	0	0	/	/	R	/
38	UP2	2	2	/	0	0	0	0	1	0	/	3	A1	/
39	UP2	2	3	35	0	0	0	0	1	0	38	4	A1	/
40	UP2	2	4	36	0	0	0	0	1	0	39	6	A2	/
41	UP2	2	4	36	0	0	0	0	1	0	40	7	A3	/
42	UP2	4	6	36	0	0	0	0	1	0	41	11	A3	/
43	UP2	2	4	36	0	0	0	0	1	0	40	9	A4	/
44	UP2	4	6	36	0	0	0	0	1	0	43	12	A4	/
45	UP2	2	2	/	0	0	0	0	1	0	/	16	A5	/
46	UP2	2	2	/	0	0	0	0	1	0	/	19	G1	/
47	UP2	2	2	/	0	0	0	0	1	0	46	21	G2	/
48	UP2	2	2	/	0	0	0	0	1	0	46	32	G4	/
49	UP2	4	6	36	0	0	0	0	1	0	46, 50	30	G3	/
50	UP2	2	4	36	0	0	0	0	1	0	47	29	G3	/
51	UP2	4	6	36	0	0	0	0	1	0	42, 44, 45	12	A4	G3
52	UP4	4	6	36	0	0	0	0	0	2	/	10	A2	/
53	UP4	4	6	36	0	0	0	0	0	2	52	11	A3	/
54	UP4	4	6	36	0	0	0	0	0	2	52	12	A4	/
55	UP4	4	6	36	0	0	0	0	0	2	/	30	G3	/
56	UP4	4	6	36	0	0	0	0	0	1	53, 54, 55	12	A4	G3

B

Example Schedule

An example of a schedule constructed by the FPTSAP model is given in Table B.1. The table visualizes the planning per go. In this schedule, we can see that pilot 16-F2-4 is assigned to mission 1 during the AM go of day 1. In the scenario used, a maximum of 8 aircraft is available per week. If less than 8 aircraft are available for a given week, the schedule is completed with na. If an aircraft was available, but not used for scheduling, the schedule displays Empty AC.

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM1	1: 16-F2-4	1: 22-SP-1	1: 23-SP-2	1: 2-IP-2	1: 7-IP-7	1: 8-F4-1	2: 18-F2-6	2: 19-WM-1
PM1	1: 5-IP-5	1: 6-IP-6	2: 22-SP-1	2: 23-SP-2	2: 2-IP-2	2: 3-IP-3	3: 18-F2-6	3: 19-WM-1
AM2	1: 13-F2-1	1: 1-IP-1	2: 12-F4-5	2: 14-F2-2	38: 19-WM-1	38: 20-WM-2	38: 2-IP-2	38: 7-IP-7
PM2	1: 15-F2-3	1: 9-F4-2	35: 23-SP-2	35: 7-IP-7	39: 20-WM-2	39: 5-IP-5	4: 10-F4-3	4: 1-IP-1
AM3	2: 4-IP-4	2: 7-IP-7	2: 8-F4-1	2: 9-F4-2	3: 1-IP-1	3: 22-SP-1	3: 23-SP-2	3: 3-IP-3
PM3	2: 20-WM-2	2: 5-IP-5	35: 1-IP-1	35: 23-SP-2	39: 19-WM-1	39: 3-IP-3	4: 22-SP-1	4: 7-IP-7
AM4	1: 10-F4-3	1: 11-F4-4	1: 12-F4-5	1: 14-F2-2	1: 20-WM-2	1: 4-IP-4	2: 16-F2-4	2: 21-WM-3
PM4	1: 18-F2-6	1: 3-IP-3	2: 10-F4-3	2: 11-F4-4	2: 13-F2-1	2: 6-IP-6	3: 16-F2-4	3: 21-WM-3
AM5	35: 11-F4-4	35: 19-WM-1	3: 17-F2-5	3: 20-WM-2	4: 23-SP-2	4: 6-IP-6	5: 1-IP-1	5: 22-SP-1
PM5	1: 17-F2-5	1: 21-WM-3	2: 15-F2-3	2: 1-IP-1	35: 6-IP-6	5: 23-SP-2	5: 2-IP-2	Empty AC
AM6	18: 14-F2-2	18: 16-F2-4	18: 18-F2-6	18: 2-IP-2	46: 20-WM-2	46: 6-IP-6	na	na
PM6	18: 15-F2-3	18: 20-WM-2	19: 19-WM-1	19: 23-SP-2	19: 4-IP-4	19: 5-IP-5	na	na
AM7	18: 18-F2-6	18: 1-IP-1	18: 20-WM-2	18: 6-IP-6	19: 15-F2-3	19: 17-F2-5	na	na
PM7	18: 10-F4-3	18: 12-F4-5	18: 13-F2-1	18: 17-F2-5	19: 18-F2-6	19: 1-IP-1	na	na
AM8	18: 19-WM-1	18: 7-IP-7	19: 10-F4-3	19: 11-F4-4	19: 14-F2-2	19: 9-F4-2	na	na
PM8	18: 22-SP-1	18: 4-IP-4	19: 20-WM-2	19: 3-IP-3	46: 19-WM-1	46: 7-IP-7	na	na
AM9	19: 12-F4-5	19: 22-SP-1	19: 2-IP-2	19: 8-F4-1	Empty AC	Empty AC	na	na
PM9	18: 11-F4-4	18: 21-WM-3	18: 8-F4-1	18: 9-F4-2	19: 13-F2-1	19: 16-F2-4	na	na
AM10	18: 16-F2-4	18: 17-F2-5	18: 21-WM-3	18: 5-IP-5	Empty AC	Empty AC	na	na
PM10	18: 23-SP-2	18: 3-IP-3	Empty AC	Empty AC	Empty AC	Empty AC	na	na
AM11	20: 14-F2-2	20: 8-F4-1	47: 20-WM-2	47: 7-IP-7	na	na	na	na
PM11	20: 1-IP-1	20: 22-SP-1	20: 23-SP-2	20: 7-IP-7	na	na	na	na
AM12	21: 22-SP-1	21: 3-IP-3	22: 23-SP-2	22: 5-IP-5	na	na	na	na
PM12	20: 3-IP-3	20: 4-IP-4	47: 19-WM-1	47: 5-IP-5	na	na	na	na
AM13	21: 23-SP-2	21: 6-IP-6	22: 22-SP-1	22: 7-IP-7	na	na	na	na
PM13	23: 23-SP-2	23: 2-IP-2	24: 22-SP-1	24: 7-IP-7	na	na	na	na
AM14	20: 15-F2-3	20: 7-IP-7	24: 1-IP-1	24: 23-SP-2	na	na	na	na
PM14	20: 10-F4-3	20: 2-IP-2	21: 11-F4-4	21: 1-IP-1	na	na	na	na

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM15	20: 11-F4-4	20: 3-IP-3	23: 1-IP-1	23: 22-SP-1	na	na	na	na
PM15	20: 12-F4-5	20: 8-F4-1	21: 10-F4-3	21: 4-IP-4	na	na	na	na
AM16	28: 23-SP-2	28: 3-IP-3	36: 16-F2-4	36: 5-IP-5	50: 20-WM-2	50: 6-IP-6	na	na
PM16	25: 23-SP-2	25: 2-IP-2	28: 1-IP-1	28: 22-SP-1	Empty AC	Empty AC	na	na
AM17	25: 22-SP-1	25: 2-IP-2	36: 18-F2-6	36: 5-IP-5	50: 19-WM-1	50: 4-IP-4	na	na
PM17	36: 14-F2-2	36: 19-WM-1	49: 20-WM-2	49: 2-IP-2	49: 4-IP-4	49: 9-F4-2	na	na
AM18	26: 22-SP-1	26: 3-IP-3	29: 23-SP-2	29: 5-IP-5	36: 10-F4-3	36: 9-F4-2	na	na
PM18	26: 1-IP-1	26: 23-SP-2	29: 22-SP-1	29: 6-IP-6	36: 17-F2-5	36: 8-F4-1	na	na
AM19	30: 1-IP-1	30: 22-SP-1	30: 23-SP-2	30: 3-IP-3	36: 12-F4-5	36: 5-IP-5	na	na
PM19	36: 10-F4-3	36: 5-IP-5	49: 14-F2-2	49: 19-WM-1	49: 1-IP-1	49: 3-IP-3	na	na
AM20	36: 20-WM-2	36: 2-IP-2	55: 13-F2-1	55: 5-IP-5	55: 8-F4-1	55: 9-F4-2	na	na
PM20	36: 16-F2-4	36: 20-WM-2	55: 12-F4-5	55: 13-F2-1	55: 3-IP-3	55: 5-IP-5	na	na
AM21	na	na	na	na	na	na	na	na
PM21	na	na	na	na	na	na	na	na
AM22	na	na	na	na	na	na	na	na
PM22	na	na	na	na	na	na	na	na
AM23	na	na	na	na	na	na	na	na
PM23	na	na	na	na	na	na	na	na
AM24	na	na	na	na	na	na	na	na
PM24	na	na	na	na	na	na	na	na
AM25	na	na	na	na	na	na	na	na
PM25	na	na	na	na	na	na	na	na
AM26	20: 14-F2-2	20: 15-F2-3	20: 17-F2-5	20: 18-F2-6	20: 6-IP-6	20: 9-F4-2	21: 12-F4-5	21: 8-F4-1
PM26	20: 12-F4-5	20: 13-F2-1	20: 15-F2-3	20: 17-F2-5	20: 18-F2-6	20: 19-WM-1	20: 1-IP-1	20: 21-WM-3
AM27	20: 13-F2-1	20: 16-F2-4	20: 17-F2-5	20: 19-WM-1	20: 20-WM-2	20: 8-F4-1	Empty AC	Empty AC
PM27	20: 16-F2-4	20: 19-WM-1	20: 20-WM-2	20: 5-IP-5	Empty AC	Empty AC	Empty AC	Empty AC
AM28	20: 10-F4-3	20: 11-F4-4	20: 16-F2-4	20: 18-F2-6	20: 19-WM-1	20: 2-IP-2	20: 4-IP-4	20: 5-IP-5
PM28	20: 12-F4-5	20: 18-F2-6	20: 19-WM-1	20: 20-WM-2	20: 21-WM-3	20: 2-IP-2	20: 5-IP-5	20: 9-F4-2
AM29	20: 11-F4-4	20: 13-F2-1	20: 16-F2-4	20: 17-F2-5	20: 20-WM-2	20: 21-WM-3	21: 15-F2-3	21: 2-IP-2
PM29	20: 10-F4-3	20: 17-F2-5	20: 1-IP-1	20: 21-WM-3	20: 6-IP-6	20: 7-IP-7	21: 14-F2-2	21: 20-WM-2
AM30	20: 14-F2-2	20: 18-F2-6	20: 3-IP-3	20: 4-IP-4	20: 6-IP-6	20: 9-F4-2	Empty AC	Empty AC
PM30	20: 16-F2-4	20: 20-WM-2	21: 13-F2-1	21: 17-F2-5	21: 18-F2-6	21: 19-WM-1	21: 21-WM-3	21: 9-F4-2
AM31	16: 10-F4-3	16: 8-F4-1	45: 19-WM-1	45: 1-IP-1	na	na	na	na
PM31	Empty AC	Empty AC	Empty AC	Empty AC	na	na	na	na
AM32	16: 1-IP-1	16: 22-SP-1	16: 23-SP-2	16: 2-IP-2	na	na	na	na
PM32	16: 10-F4-3	16: 11-F4-4	17: 8-F4-1	17: 9-F4-2	na	na	na	na
AM33	Empty AC	Empty AC	Empty AC	Empty AC	na	na	na	na
PM33	Empty AC	Empty AC	Empty AC	Empty AC	na	na	na	na
AM34	Empty AC	Empty AC	Empty AC	Empty AC	na	na	na	na
PM34	17: 22-SP-1	17: 23-SP-2	17: 5-IP-5	17: 6-IP-6	na	na	na	na
AM35	Empty AC	Empty AC	Empty AC	Empty AC	na	na	na	na
PM35	45: 1-IP-1	45: 20-WM-2	Empty AC	Empty AC	na	na	na	na
AM36	35: 2-IP-2	35: 7-IP-7	6: 11-F4-4	6: 12-F4-5	6: 23-SP-2	6: 6-IP-6	na	na
PM36	35: 11-F4-4	35: 2-IP-2	6: 10-F4-3	6: 15-F2-3	6: 22-SP-1	6: 4-IP-4	na	na
AM37	36: 15-F2-3	36: 17-F2-5	40: 19-WM-1	40: 1-IP-1	Empty AC	Empty AC	na	na
PM37	36: 19-WM-1	36: 7-IP-7	52: 10-F4-3	52: 13-F2-1	52: 3-IP-3	52: 5-IP-5	na	na
AM38	36: 17-F2-5	36: 7-IP-7	52: 12-F4-5	52: 13-F2-1	52: 3-IP-3	52: 6-IP-6	na	na
PM38	36: 2-IP-2	36: 8-F4-1	40: 20-WM-2	40: 6-IP-6	Empty AC	Empty AC	na	na

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM39	10: 15-F2-3	10: 2-IP-2	10: 4-IP-4	10: 6-IP-6	36: 13-F2-1	36: 17-F2-5	na	na
PM39	10: 11-F4-4	10: 23-SP-2	10: 2-IP-2	10: 4-IP-4	36: 10-F4-3	36: 12-F4-5	na	na
AM40	10: 10-F4-3	10: 15-F2-3	10: 22-SP-1	10: 5-IP-5	36: 2-IP-2	36: 4-IP-4	na	na
PM40	13: 22-SP-1	13: 23-SP-2	13: 2-IP-2	13: 3-IP-3	35: 11-F4-4	35: 18-F2-6	na	na
AM41	29: 11-F4-4	29: 15-F2-3	36: 1-IP-1	36: 23-SP-2	na	na	na	na
PM41	27: 1-IP-1	27: 22-SP-1	27: 23-SP-2	27: 7-IP-7	na	na	na	na
AM42	29: 13-F2-1	29: 3-IP-3	36: 2-IP-2	36: 7-IP-7	na	na	na	na
PM42	29: 13-F2-1	29: 1-IP-1	36: 7-IP-7	36: 8-F4-1	na	na	na	na
AM43	29: 12-F4-5	29: 1-IP-1	36: 14-F2-2	36: 18-F2-6	na	na	na	na
PM43	29: 14-F2-2	29: 1-IP-1	36: 3-IP-3	36: 7-IP-7	na	na	na	na
AM44	29: 6-IP-6	29: 9-F4-2	36: 14-F2-2	36: 1-IP-1	na	na	na	na
PM44	29: 14-F2-2	29: 15-F2-3	36: 13-F2-1	36: 23-SP-2	na	na	na	na
AM45	29: 11-F4-4	29: 12-F4-5	36: 20-WM-2	36: 4-IP-4	na	na	na	na
PM45	29: 13-F2-1	29: 15-F2-3	36: 1-IP-1	36: 21-WM-3	na	na	na	na
AM46	na	na	na	na	na	na	na	na
PM46	na	na	na	na	na	na	na	na
AM47	na	na	na	na	na	na	na	na
PM47	na	na	na	na	na	na	na	na
AM48	na	na	na	na	na	na	na	na
PM48	na	na	na	na	na	na	na	na
AM49	na	na	na	na	na	na	na	na
PM49	na	na	na	na	na	na	na	na
AM50	na	na	na	na	na	na	na	na
PM50	na	na	na	na	na	na	na	na
AM51	35: 1-IP-1	35: 3-IP-3	6: 10-F4-3	6: 14-F2-2	6: 2-IP-2	6: 8-F4-1	Empty AC	Empty AC
PM51	35: 10-F4-3	35: 1-IP-1	6: 12-F4-5	6: 5-IP-5	6: 8-F4-1	6: 9-F4-2	Empty AC	Empty AC
AM52	35: 1-IP-1	35: 2-IP-2	6: 11-F4-4	6: 13-F2-1	6: 14-F2-2	6: 3-IP-3	Empty AC	Empty AC
PM52	35: 1-IP-1	35: 22-SP-1	6: 10-F4-3	6: 13-F2-1	6: 3-IP-3	6: 4-IP-4	Empty AC	Empty AC
AM53	10: 12-F4-5	10: 14-F2-2	10: 1-IP-1	10: 9-F4-2	36: 17-F2-5	36: 18-F2-6	Empty AC	Empty AC
PM53	35: 21-WM-3	35: 23-SP-2	6: 15-F2-3	6: 2-IP-2	6: 5-IP-5	6: 9-F4-2	Empty AC	Empty AC
AM54	35: 15-F2-3	35: 5-IP-5	6: 13-F2-1	6: 14-F2-2	6: 1-IP-1	6: 6-IP-6	Empty AC	Empty AC
PM54	35: 10-F4-3	35: 22-SP-1	6: 11-F4-4	6: 12-F4-5	6: 3-IP-3	6: 8-F4-1	Empty AC	Empty AC
AM55	35: 17-F2-5	35: 19-WM-1	6: 15-F2-3	6: 1-IP-1	6: 7-IP-7	6: 9-F4-2	Empty AC	Empty AC
PM55	10: 11-F4-4	10: 13-F2-1	10: 1-IP-1	10: 9-F4-2	36: 15-F2-3	36: 4-IP-4	Empty AC	Empty AC
AM56	36: 21-WM-3	36: 6-IP-6	54: 11-F4-4	54: 12-F4-5	54: 13-F2-1	54: 1-IP-1	na	na
PM56	36: 19-WM-1	36: 3-IP-3	43: 20-WM-2	43: 7-IP-7	Empty AC	Empty AC	na	na
AM57	36: 10-F4-3	36: 17-F2-5	9: 1-IP-1	9: 23-SP-2	Empty AC	Empty AC	na	na
PM57	36: 12-F4-5	36: 16-F2-4	54: 13-F2-1	54: 5-IP-5	54: 6-IP-6	54: 9-F4-2	na	na
AM58	36: 10-F4-3	36: 14-F2-2	9: 22-SP-1	9: 3-IP-3	Empty AC	Empty AC	na	na
PM58	12: 22-SP-1	12: 23-SP-2	12: 2-IP-2	12: 6-IP-6	36: 15-F2-3	36: 21-WM-3	na	na
AM59	36: 1-IP-1	36: 23-SP-2	43: 19-WM-1	43: 5-IP-5	Empty AC	Empty AC	na	na
PM59	36: 12-F4-5	36: 13-F2-1	44: 11-F4-4	44: 15-F2-3	44: 20-WM-2	44: 2-IP-2	na	na
AM60	15: 22-SP-1	15: 23-SP-2	15: 6-IP-6	15: 7-IP-7	36: 16-F2-4	36: 9-F4-2	na	na
PM60	36: 15-F2-3	36: 18-F2-6	44: 13-F2-1	44: 19-WM-1	44: 1-IP-1	44: 7-IP-7	na	na
AM61	29: 7-IP-7	29: 8-F4-1	36: 13-F2-1	36: 6-IP-6	na	na	na	na
PM61	29: 11-F4-4	29: 8-F4-1	36: 15-F2-3	36: 1-IP-1	na	na	na	na
AM62	29: 10-F4-3	29: 9-F4-2	36: 1-IP-1	36: 6-IP-6	na	na	na	na
PM62	29: 2-IP-2	29: 9-F4-2	36: 13-F2-1	36: 15-F2-3	na	na	na	na

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM63	29: 14-F2-2	29: 2-IP-2	36: 1-IP-1	36: 23-SP-2	na	na	na	na
PM63	29: 2-IP-2	29: 3-IP-3	36: 20-WM-2	36: 8-F4-1	na	na	na	na
AM64	29: 3-IP-3	29: 4-IP-4	36: 13-F2-1	36: 1-IP-1	na	na	na	na
PM64	29: 10-F4-3	29: 12-F4-5	36: 13-F2-1	36: 6-IP-6	na	na	na	na
AM65	29: 4-IP-4	29: 5-IP-5	36: 13-F2-1	36: 15-F2-3	na	na	na	na
PM65	29: 5-IP-5	29: 8-F4-1	36: 13-F2-1	36: 6-IP-6	na	na	na	na
AM66	35: 21-WM-3	35: 22-SP-1	6: 17-F2-5	6: 19-WM-1	6: 20-WM-2	6: 7-IP-7	Empty AC	Empty AC
PM66	35: 19-WM-1	35: 22-SP-1	6: 16-F2-4	6: 20-WM-2	6: 21-WM-3	6: 5-IP-5	Empty AC	Empty AC
AM67	10: 13-F2-1	10: 19-WM-1	10: 20-WM-2	10: 7-IP-7	36: 2-IP-2	36: 6-IP-6	Empty AC	Empty AC
PM67	35: 8-F4-1	35: 9-F4-2	6: 17-F2-5	6: 18-F2-6	6: 19-WM-1	6: 21-WM-3	Empty AC	Empty AC
AM68	10: 14-F2-2	10: 18-F2-6	10: 19-WM-1	10: 8-F4-1	36: 11-F4-4	36: 4-IP-4	Empty AC	Empty AC
PM68	10: 16-F2-4	10: 20-WM-2	10: 21-WM-3	10: 7-IP-7	36: 15-F2-3	36: 1-IP-1	Empty AC	Empty AC
AM69	10: 16-F2-4	10: 18-F2-6	10: 19-WM-1	10: 8-F4-1	36: 13-F2-1	36: 1-IP-1	Empty AC	Empty AC
PM69	35: 12-F4-5	35: 22-SP-1	6: 16-F2-4	6: 18-F2-6	6: 19-WM-1	6: 7-IP-7	Empty AC	Empty AC
AM70	13: 10-F4-3	13: 16-F2-4	35: 2-IP-2	35: 8-F4-1	6: 17-F2-5	6: 21-WM-3	Empty AC	Empty AC
PM70	35: 15-F2-3	35: 1-IP-1	6: 16-F2-4	6: 20-WM-2	6: 2-IP-2	6: 4-IP-4	Empty AC	Empty AC
AM71	36: 18-F2-6	36: 6-IP-6	53: 10-F4-3	53: 12-F4-5	53: 13-F2-1	53: 4-IP-4	na	na
PM71	36: 16-F2-4	36: 1-IP-1	7: 22-SP-1	7: 2-IP-2	Empty AC	Empty AC	na	na
AM72	36: 14-F2-2	36: 17-F2-5	41: 1-IP-1	41: 20-WM-2	Empty AC	Empty AC	na	na
PM72	11: 22-SP-1	11: 23-SP-2	11: 3-IP-3	11: 7-IP-7	36: 14-F2-2	36: 9-F4-2	na	na
AM73	36: 15-F2-3	36: 17-F2-5	7: 23-SP-2	7: 4-IP-4	Empty AC	Empty AC	na	na
PM73	36: 3-IP-3	36: 6-IP-6	41: 19-WM-1	41: 1-IP-1	Empty AC	Empty AC	na	na
AM74	36: 3-IP-3	36: 6-IP-6	42: 11-F4-4	42: 20-WM-2	42: 2-IP-2	42: 7-IP-7	na	na
PM74	36: 11-F4-4	36: 13-F2-1	42: 12-F4-5	42: 19-WM-1	42: 2-IP-2	42: 4-IP-4	na	na
AM75	36: 11-F4-4	36: 6-IP-6	53: 13-F2-1	53: 3-IP-3	53: 5-IP-5	53: 8-F4-1	na	na
PM75	14: 22-SP-1	14: 23-SP-2	14: 2-IP-2	14: 6-IP-6	36: 17-F2-5	36: 1-IP-1	na	na
AM76	35: 2-IP-2	4: 14-F2-2	4: 4-IP-4	Empty AC	Empty AC	Empty AC	na	na
PM76	35: 23-SP-2	5: 14-F2-2	5: 9-F4-2	Empty AC	Empty AC	Empty AC	na	na
AM77	Empty AC	Empty AC	Empty AC	Empty AC	Empty AC	Empty AC	na	na
PM77	35: 22-SP-1	35: 23-SP-2	4: 16-F2-4	4: 2-IP-2	5: 17-F2-5	5: 21-WM-3	na	na
AM78	35: 22-SP-1	35: 23-SP-2	4: 11-F4-4	4: 15-F2-3	5: 13-F2-1	5: 18-F2-6	na	na
PM78	35: 22-SP-1	35: 23-SP-2	4: 20-WM-2	4: 8-F4-1	5: 3-IP-3	5: 5-IP-5	na	na
AM79	35: 22-SP-1	35: 23-SP-2	4: 13-F2-1	4: 18-F2-6	5: 19-WM-1	5: 4-IP-4	na	na
PM79	35: 13-F2-1	35: 3-IP-3	4: 17-F2-5	4: 9-F4-2	5: 11-F4-4	5: 8-F4-1	na	na
AM80	35: 1-IP-1	35: 23-SP-2	4: 12-F4-5	4: 19-WM-1	5: 6-IP-6	5: 7-IP-7	na	na
PM80	35: 1-IP-1	35: 23-SP-2	5: 10-F4-3	5: 12-F4-5	5: 15-F2-3	5: 16-F2-4	na	na
AM81	11: 14-F2-2	11: 17-F2-5	11: 21-WM-3	11: 6-IP-6	36: 10-F4-3	36: 23-SP-2	na	na
PM81	11: 15-F2-3	11: 19-WM-1	11: 1-IP-1	11: 20-WM-2	36: 16-F2-4	36: 6-IP-6	na	na
AM82	36: 14-F2-2	36: 7-IP-7	7: 11-F4-4	7: 3-IP-3	Empty AC	Empty AC	na	na
PM82	11: 11-F4-4	11: 19-WM-1	11: 20-WM-2	11: 6-IP-6	36: 10-F4-3	36: 5-IP-5	na	na
AM83	11: 16-F2-4	11: 18-F2-6	11: 21-WM-3	11: 5-IP-5	36: 12-F4-5	36: 22-SP-1	na	na
PM83	11: 13-F2-1	11: 17-F2-5	11: 18-F2-6	11: 9-F4-2	36: 7-IP-7	36: 8-F4-1	na	na
AM84	36: 18-F2-6	36: 7-IP-7	7: 11-F4-4	7: 8-F4-1	Empty AC	Empty AC	na	na
PM84	35: 18-F2-6	35: 20-WM-2	8: 22-SP-1	8: 23-SP-2	8: 6-IP-6	8: 7-IP-7	na	na
AM85	11: 15-F2-3	11: 1-IP-1	11: 8-F4-1	11: 9-F4-2	36: 10-F4-3	36: 20-WM-2	na	na
PM85	11: 10-F4-3	11: 13-F2-1	11: 14-F2-2	11: 16-F2-4	36: 15-F2-3	36: 5-IP-5	na	na
AM86	12: 20-WM-2	12: 21-WM-3	12: 3-IP-3	12: 8-F4-1	36: 18-F2-6	36: 23-SP-2	na	na
PM86	12: 16-F2-4	12: 18-F2-6	12: 19-WM-1	12: 7-IP-7	36: 11-F4-4	36: 23-SP-2	na	na

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM87	36: 11-F4-4	36: 22-SP-1	56: 10-F4-3	56: 13-F2-1	56: 17-F2-5	56: 5-IP-5	na	na
PM87	12: 10-F4-3	12: 14-F2-2	12: 20-WM-2	12: 9-F4-2	36: 23-SP-2	36: 7-IP-7	na	na
AM88	12: 12-F4-5	12: 13-F2-1	12: 19-WM-1	12: 8-F4-1	36: 17-F2-5	36: 22-SP-1	na	na
PM88	34: 7-IP-7	34: 9-F4-2	36: 19-WM-1	36: 1-IP-1	Empty AC	Empty AC	na	na
AM89	36: 13-F2-1	36: 16-F2-4	9: 14-F2-2	9: 8-F4-1	Empty AC	Empty AC	na	na
PM89	12: 15-F2-3	12: 18-F2-6	12: 3-IP-3	12: 4-IP-4	36: 13-F2-1	36: 16-F2-4	na	na
AM90	36: 12-F4-5	36: 8-F4-1	9: 13-F2-1	9: 9-F4-2	Empty AC	Empty AC	na	na
PM90	12: 14-F2-2	12: 16-F2-4	12: 21-WM-3	12: 4-IP-4	36: 13-F2-1	36: 20-WM-2	na	na
AM91	30: 10-F4-3	30: 13-F2-1	30: 14-F2-2	30: 15-F2-3	36: 1-IP-1	36: 23-SP-2	na	na
PM91	30: 10-F4-3	30: 12-F4-5	30: 19-WM-1	30: 9-F4-2	36: 1-IP-1	36: 22-SP-1	na	na
AM92	30: 11-F4-4	30: 21-WM-3	30: 2-IP-2	30: 4-IP-4	36: 23-SP-2	36: 5-IP-5	na	na
PM92	30: 11-F4-4	30: 14-F2-2	30: 15-F2-3	30: 8-F4-1	36: 23-SP-2	36: 3-IP-3	na	na
AM93	30: 13-F2-1	30: 18-F2-6	30: 19-WM-1	30: 6-IP-6	36: 17-F2-5	36: 20-WM-2	na	na
PM93	30: 10-F4-3	30: 17-F2-5	30: 19-WM-1	30: 2-IP-2	36: 18-F2-6	36: 20-WM-2	na	na
AM94	30: 20-WM-2	30: 21-WM-3	30: 4-IP-4	30: 6-IP-6	36: 1-IP-1	36: 3-IP-3	na	na
PM94	30: 11-F4-4	30: 6-IP-6	30: 7-IP-7	30: 8-F4-1	36: 1-IP-1	36: 3-IP-3	na	na
AM95	30: 12-F4-5	30: 13-F2-1	30: 15-F2-3	30: 7-IP-7	36: 22-SP-1	36: 3-IP-3	na	na
PM95	30: 1-IP-1	30: 20-WM-2	30: 5-IP-5	30: 7-IP-7	36: 22-SP-1	36: 3-IP-3	na	na
AM96	36: 7-IP-7	36: 9-F4-2	9: 2-IP-2	9: 6-IP-6	Empty AC	Empty AC	na	na
PM96	34: 14-F2-2	34: 2-IP-2	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
AM97	36: 7-IP-7	36: 9-F4-2	9: 11-F4-4	9: 18-F2-6	Empty AC	Empty AC	na	na
PM97	34: 10-F4-3	34: 8-F4-1	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
AM98	34: 11-F4-4	34: 4-IP-4	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
PM98	34: 5-IP-5	34: 6-IP-6	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
AM99	34: 12-F4-5	34: 1-IP-1	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
PM99	36: 7-IP-7	36: 9-F4-2	9: 10-F4-3	9: 12-F4-5	Empty AC	Empty AC	na	na
AM100	36: 7-IP-7	36: 9-F4-2	9: 15-F2-3	9: 4-IP-4	Empty AC	Empty AC	na	na
PM100	34: 15-F2-3	34: 3-IP-3	36: 7-IP-7	36: 9-F4-2	Empty AC	Empty AC	na	na
AM101	34: 17-F2-5	34: 20-WM-2	36: 10-F4-3	36: 19-WM-1	Empty AC	Empty AC	na	na
PM101	36: 18-F2-6	36: 8-F4-1	9: 17-F2-5	9: 20-WM-2	Empty AC	Empty AC	na	na
AM102	36: 10-F4-3	36: 5-IP-5	9: 16-F2-4	9: 21-WM-3	Empty AC	Empty AC	na	na
PM102	34: 18-F2-6	34: 19-WM-1	36: 17-F2-5	36: 23-SP-2	Empty AC	Empty AC	na	na
AM103	36: 11-F4-4	36: 19-WM-1	9: 17-F2-5	9: 18-F2-6	Empty AC	Empty AC	na	na
PM103	34: 16-F2-4	34: 18-F2-6	36: 14-F2-2	36: 9-F4-2	Empty AC	Empty AC	na	na
AM104	34: 16-F2-4	34: 20-WM-2	36: 12-F4-5	36: 4-IP-4	Empty AC	Empty AC	na	na
PM104	36: 20-WM-2	36: 5-IP-5	9: 16-F2-4	9: 21-WM-3	Empty AC	Empty AC	na	na
AM105	34: 17-F2-5	34: 21-WM-3	36: 16-F2-4	36: 2-IP-2	Empty AC	Empty AC	na	na
PM105	34: 13-F2-1	34: 21-WM-3	36: 10-F4-3	36: 1-IP-1	Empty AC	Empty AC	na	na
AM106	36: 17-F2-5	36: 1-IP-1	7: 14-F2-2	7: 8-F4-1	Empty AC	Empty AC	na	na
PM106	36: 21-WM-3	36: 8-F4-1	7: 10-F4-3	7: 6-IP-6	Empty AC	Empty AC	na	na
AM107	36: 1-IP-1	36: 20-WM-2	7: 13-F2-1	7: 15-F2-3	Empty AC	Empty AC	na	na
PM107	36: 1-IP-1	36: 8-F4-1	7: 15-F2-3	7: 9-F4-2	Empty AC	Empty AC	na	na
AM108	36: 22-SP-1	36: 8-F4-1	7: 2-IP-2	7: 9-F4-2	Empty AC	Empty AC	na	na
PM108	36: 11-F4-4	36: 8-F4-1	7: 10-F4-3	7: 5-IP-5	Empty AC	Empty AC	na	na
AM109	36: 12-F4-5	36: 14-F2-2	7: 6-IP-6	7: 7-IP-7	Empty AC	Empty AC	na	na
PM109	36: 11-F4-4	36: 22-SP-1	7: 13-F2-1	7: 14-F2-2	Empty AC	Empty AC	na	na
AM110	36: 17-F2-5	36: 9-F4-2	7: 12-F4-5	7: 3-IP-3	Empty AC	Empty AC	na	na
PM110	36: 10-F4-3	36: 8-F4-1	7: 12-F4-5	7: 4-IP-4	Empty AC	Empty AC	na	na

Table B.1: Example of a schedule output by the FPTSAP model.

	Aircraft							
	1	2	3	4	5	6	7	8
AM111	36: 1-IP-1	36: 23-SP-2	7: 18-F2-6	7: 19-WM-1	na	na	na	na
PM111	36: 21-WM-3	36: 7-IP-7	7: 17-F2-5	7: 20-WM-2	na	na	na	na
AM112	36: 15-F2-3	36: 1-IP-1	7: 16-F2-4	7: 17-F2-5	na	na	na	na
PM112	36: 17-F2-5	36: 19-WM-1	7: 16-F2-4	7: 20-WM-2	na	na	na	na
AM113	36: 21-WM-3	36: 8-F4-1	7: 16-F2-4	7: 20-WM-2	na	na	na	na
PM113	36: 14-F2-2	36: 3-IP-3	7: 21-WM-3	7: 5-IP-5	na	na	na	na
AM114	36: 18-F2-6	36: 9-F4-2	7: 19-WM-1	7: 7-IP-7	na	na	na	na
PM114	36: 1-IP-1	36: 7-IP-7	7: 18-F2-6	7: 19-WM-1	na	na	na	na
AM115	36: 23-SP-2	36: 2-IP-2	7: 17-F2-5	7: 21-WM-3	na	na	na	na
PM115	36: 23-SP-2	36: 2-IP-2	7: 18-F2-6	7: 21-WM-3	na	na	na	na
AM116	32: 11-F4-4	32: 12-F4-5	32: 14-F2-2	32: 16-F2-4	32: 17-F2-5	32: 1-IP-1	32: 20-WM-2	32: 21-WM-3
PM116	32: 10-F4-3	32: 11-F4-4	32: 13-F2-1	32: 15-F2-3	32: 16-F2-4	32: 17-F2-5	32: 1-IP-1	32: 8-F4-1
AM117	32: 12-F4-5	32: 16-F2-4	32: 20-WM-2	32: 21-WM-3	32: 3-IP-3	32: 4-IP-4	32: 6-IP-6	32: 9-F4-2
PM117	31: 22-SP-1	31: 4-IP-4	32: 17-F2-5	32: 19-WM-1	32: 21-WM-3	32: 8-F4-1	48: 20-WM-2	48: 6-IP-6
AM118	31: 23-SP-2	31: 3-IP-3	32: 13-F2-1	32: 14-F2-2	32: 18-F2-6	32: 5-IP-5	32: 7-IP-7	32: 8-F4-1
PM118	32: 13-F2-1	32: 15-F2-3	32: 20-WM-2	32: 22-SP-1	32: 23-SP-2	32: 3-IP-3	32: 4-IP-4	32: 7-IP-7
AM119	32: 10-F4-3	32: 18-F2-6	32: 19-WM-1	32: 1-IP-1	32: 5-IP-5	32: 7-IP-7	33: 23-SP-2	33: 3-IP-3
PM119	32: 10-F4-3	32: 18-F2-6	32: 2-IP-2	32: 4-IP-4	32: 5-IP-5	32: 9-F4-2	33: 1-IP-1	33: 22-SP-1
AM120	31: 13-F2-1	31: 14-F2-2	31: 16-F2-4	31: 18-F2-6	32: 11-F4-4	32: 12-F4-5	32: 15-F2-3	32: 2-IP-2
PM120	32: 14-F2-2	32: 2-IP-2	32: 3-IP-3	32: 9-F4-2	48: 19-WM-1	48: 6-IP-6	Empty AC	Empty AC
AM121	36: 10-F4-3	36: 13-F2-1	51: 12-F4-5	51: 19-WM-1	51: 1-IP-1	51: 6-IP-6	na	na
PM121	29: 16-F2-4	29: 20-WM-2	36: 12-F4-5	36: 13-F2-1	Empty AC	Empty AC	na	na
AM122	29: 18-F2-6	29: 7-IP-7	36: 11-F4-4	36: 15-F2-3	Empty AC	Empty AC	na	na
PM122	29: 10-F4-3	29: 16-F2-4	36: 5-IP-5	36: 6-IP-6	Empty AC	Empty AC	na	na
AM123	29: 17-F2-5	29: 21-WM-3	36: 12-F4-5	36: 22-SP-1	Empty AC	Empty AC	na	na
PM123	36: 17-F2-5	36: 1-IP-1	51: 15-F2-3	51: 20-WM-2	51: 4-IP-4	51: 7-IP-7	na	na
AM124	29: 16-F2-4	29: 20-WM-2	36: 14-F2-2	36: 4-IP-4	Empty AC	Empty AC	na	na
PM124	29: 17-F2-5	29: 7-IP-7	36: 15-F2-3	36: 22-SP-1	Empty AC	Empty AC	na	na
AM125	29: 18-F2-6	29: 20-WM-2	36: 15-F2-3	36: 22-SP-1	Empty AC	Empty AC	na	na
PM125	29: 17-F2-5	29: 21-WM-3	36: 20-WM-2	36: 2-IP-2	Empty AC	Empty AC	na	na

Bibliography

[1] Davut Aslan. A decision support system for effective scheduling in an f-16 pilot training squadron. Master's thesis, Air Force Institute of Technology, 2003. URL https://apps.dtic.mil/sti/citations/ADA415710.

[2] Jens Clausen. Brand and bound algorithms: Principles and examples. techreport, University of Copenhagen, 1999.

[3] Spyder Project Contributors. Spyder documentation, 2021. URL https://docs.spyder-ide.org/current/index.html.

[4] Ugur Erdemir. Optimizing flight schedule by an automated decision support system. Master's thesis, Air Force Institute Of Technology, 2014. URL https://apps.dtic.mil/sti/citations/ADA601995.

[5] Python Software Foundation. Python 3.7 documentation, 2021. URL https://docs.python.org/3.7/.

[6] R. S. Garfinkel and G. L. Nemhouser. The set-partioning problem: Set covering with equality constraints. Operations Research, 17(5):848–856, 1969. doi: 10.1287/opre.17.5.848.

[7] Michel Gendreau. Handbook of metaheuristics, chapter An introduction to Tabu search, pages 37–54. Kluwer Academic Publishers, 2003.

[8] Fred Glover. Future paths for integer programming and links to artificial intelligence. Computers & Operations Research, 13(5):533–549, 1986. doi: 10.1016/0305-0548(86)90048-1.

[9] Reinout Guljé. A priority-based method for scheduling and rescheduling fighter pilot continuation training. Master's thesis, Erasmus University Rotterdam, 2018.

[10] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021. URL https://www.gurobi.com.

[11] Asa Holm. Manpower planning in airlines. Master's thesis, Linköpings Universitet, 2008. URL https://www.diva-portal.org/smash/record.jsf?pid=diva2:37669&dswid=2588.

[12] Bong-Kyun Kim and Yeong-Dae Kim. Heuristic algorithms for assigning and scheduling flight missions in a military aviation unit. Computers & Industrial Engineering, 61:1309–1317, 2011. doi: 10.1016/j.cie.2011.08.004.

[13] Niklas Kohl and Stefan E. Karisch. Airline crew rostering: Problem types, modeling and optimization. Annals of Operations Research, 127:223–257, 2004. doi: 10.1023/b:anor.0000019091.54417.ca.

[14] George Kozanidis. Optimal assignment of aircrew trainees to simulator and classroom training sessions subject to seniority and preference restrictions. Journal of Air Transport Management, 59:143–154, 2017. ISSN 0969-6997. doi: 10.1016/j.jairtraman.2016.11.012.

[15] Marta Mesquita and Ana Paias. Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. Computers & Operations Research, 35(5):1562–1575, May 2008. doi: 10.1016/j.cor.2006.09.001.

[16] Thomas M. Newlon. Mathematical programming model for fighter training squadron pilot scheduling. Master's thesis, Air Force Institue of Technology, 2007.

[17] Cuong T. Nguyen. An interactive decision support system for scheduling fighter pilot training. Master's thesis, Air Force Institute of Technology, 2002. URL https://apps.dtic.mil/sti/citations/ADA401391.

[18] Jon D. Petersen, Gustaf Sölveling, John-Paul Clarke, Ellis L. Johnson, and Sergey Shebalov. An optimization approach to airline integrated recovery. Transportation Science, 46(4):482–500, 2012. doi: 10.1287/trsc.1120.0414.

[19] Michael L. Pinedo. Scheduling. Theory, algorithms and systems. Springer, 2016. ISBN 978-3-319-26578-0. doi: 10.1007/978-3-319-26580-3.

[20] Xiangton Qi, Jonathan F. Bard, and Gang Ju. Class scheduling for pilot training. Operations Research, 52 (1):148–162, January 2004. ISSN 0030-364X. doi: 10.1287/opre.1030.0076.

[21] Ragheb Rahmaniani, Teodor Gabriel, Crainic, and Walter Rei Michel Gendreau. The benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3):801–817, jun 2017. doi: 10.1016/j.ejor.2016.12.005.

[22] Mauricio G. C. Resende and Celso C. Ribeiro. Handbook of metaheuristics, chapter Greedy randomized adaptive search procedures, pages 219–249. Kluwer Academic Publishers, 2003. ISBN 1-4020-7263-5.

[23] G. T. Ross and R. M. Soland. A branch and bound algorithm for the generalized assignment problem. Mathematical programming, 8:91–103, December 1975. doi: 10.1007/bf01580430.

[24] S. Salcedo-Sanz. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. Physics Reports, 655:1–70, 2016. doi: 10.1016/j.physrep.2016.08.001.

[25] Javier Silvente, Georgios M. Kopanos, Efstratios N. Pistikopoulos, and Antonio Espuña. A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids. Applied Energy, 155:485–501, 2015. doi: 10.1016/j.apenergy.2015.05.090.

[26] Mike Taks. Scheduling training missions and pilots. Master's thesis, Utrecht University, 2016. URL https://dspace.library.uu.nl/handle/1874/328765.

[27] Z. Caner Taskin. Wiley encyclopedia of operations research and management science, chapter Benders decomposition. John Wiley & Sons, Inc, 2010.

[28] Dirk Jan van den Berg, Michael Beltman, Bas Bijlsma, Maarten Boef, Henk Geveke, Stan Kaatee, Niels Kastelein, Gijs van Keulen, Pieter van Marken, Peter Reesink, Philip Strik, Gerbe Verhaaf, Arthur van Vliet, David van Weel, and Hans Wehren. Zicht op gereedheid. Technical report, Rijksoverheid, 2017. URL https://rijksbegroting.nl/system/files/12/ibo-gereedstelling-2017.pdf.

[29] Joey van Kempen. Robust cockpit crew training scheduling. Master's thesis, Delft University of Technology, 2019. URL http://resolver.tudelft.nl/uuid:98e15daa-2053-49d7-8d91-ea2ff3933523.

[30] Murat Yavuz. Optimizing an f-16 weekly pilot schedule for the turkish air force. Master's thesis, Air Force Institute of Technology, 2010. URL https://scholar.afit.edu/etd/2100/.

[31] Gang Yu, Julian Pachon, Benjamin Thengvall, Darryal Chandler, and Al Wilson. Optimizing pilot planning and training for continental airlines. INFORMS Journal on Applied Analytics, 34(4):253–264, July 2004. ISSN 0092-2102. doi: 10.1287/inte.1040.0082.

[32] T. Öncan. A survey of the generalized assignment problem and its applications. Information systems and operational research, 45(3):123–141, 2007. doi: 10.3138/infor.45.3.123.