Disruptions and Robustness in Air Force Crew Scheduling

by

Christopher Ho-Yen Chin

B.S., University of California, Berkeley (2018)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . .
Department of Aeronautics and Astronautics
May 18, 2021

Certified by . . . .
Hamsa Balakrishnan
Professor, Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . .
Zoltan Spakovszky
Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

Disruptions and Robustness in Air Force Crew Scheduling

by

Christopher Ho-Yen Chin

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Air Force crew scheduling involves assigning pilots to flights to fulfill mission duties and complete training requirements. Because of complex qualification requirements, as well as crew rest and availability constraints, Air Force crew scheduling is a challenging combinatorial optimization problem. Further, last-minute disruptions and uncertainties in factors like flight duration and pilot availability motivate the need for more robust schedules. Traditionally, this has been a manual, tedious, and time-consuming process. In this thesis, we leverage optimization techniques to improve the crew scheduling process. We start with a baseline integer program formulation. We develop objective functions based on two known scheduler priorities: maximizing training requirements completed, and minimizing overqualification (assigning the lowest qualified pilot feasible for each pilot seat). Then, we present a formulation to handle disruptions to an original schedule. We develop an intuitive schedule visualization tool that we use for user studies, and discuss user feedback on our scheduling algorithms. Finally, we identify key uncertainties in Air Force crew scheduling and contrast them with commercial aviation. We adapt two concepts from commercial aviation for robust crew scheduling: buffer times (slack time between two successive flights operated by the same pilots) and move-up crews (back-up crews for substitution when pilots become unavailable). This work will contribute to the core of the *Puckboard* scheduling software under development by the Air Force for crew scheduling. [1]

Thesis Supervisor: Hamsa Balakrishnan
Title: Professor, Aeronautics and Astronautics

[1] The views expressed are those of the author and do not reflect the official guidance or position of the United States Government, the Department of Defense, or of the United States Air Force.

Acknowledgments

First and foremost, I would like to thank my advisor Prof. Hamsa Balakrishnan. She leads by example, whether by asking insightful questions, communicating ideas elegantly, or staying up to push a paper past the finish line. In addition, she really cares about her students. I am so grateful for the opportunity to work with her.

I would also like to thank Matthew Koch, whom I worked with closely on the USAF-MIT AI Accelerator project. He assisted me in understanding Air Force terminology and how schedulers and pilots solve problems. I would also like to thank other members of the Accelerator team, including Siddharth Nayak, Travis Smith, Kendrick Cancio, Jessamyn Liu, Amy Alexander, Allison Chang, Ronisha Carter, and Michael Snyder.

On other projects, I have enjoyed working with Karthik Gopalakrishnan and Max Li (both of whom are graduating soon), and I look forward to continuing to work with them. I would also like to thank Husni Idris at NASA Ames who mentored me before coming to MIT and taught me how to conduct research. I am also grateful for the opportunity to work with Antony Evans and Maxim Egorov at Airbus UTM.

I wouldn't be here without my family. I would like to thank my parents for their unwavering, unconditional support and my two older sister Jennifer and Natalie for always spoiling me and being the best older sisters. Finally, I would like to thank my partner Justina for always encouraging me and bringing joy into my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Air Force crew scheduling is a complex yet largely manual process. Several considerations are taken into account when creating schedules. These considerations include pilot availability, qualification levels required, training requirements to complete, and likelihood of schedule disruption. Because of the need to consider so many factors, a typical schedule can take 3 airmen working 9 hours/day for a total of 27 person-hours to create. Thus, there is an opportunity to improve the scheduling process so that airmen can focus on their other important duties.

Because the current scheduling process is so manual and ad-hoc, it is also likely suboptimal. Air Force schedulers (hence referred to as "schedulers") that we talked to state that they frequently have to discard previously made schedules in the face of disruptions. In the Air Force, schedule disruptions are very common, due to, for example, pop-up missions, unexpected pilot unavailability, or maintenance issues. We expect that optimization techniques can not only save schedulers time, but improve the quality of schedules.

1.2 Project Background

Because the Air Force recognizes the need to improve the scheduling process, it is developing *Puckboard*, a web-based software application to assist schedulers. *Puckboard* is currently deployed for C-17 squadrons but is expected to be rolled out to other squadrons in the Air Force. One of the main successes of *Puckboard* so far has been housing data from various databases in one centralized place. The name for *Puckboard* stems from the fact that schedulers used to use pucks on whiteboards as representation of pilot assignments when creating schedules. With *Puckboard*, schedulers no longer have to cross-reference several whiteboards or Excel documents, which has already led to efficiency gains.

The goal of *Puckboard* is not to replace schedulers, but assist them. Currently, *Puckboard* is used primarily as a centralized data source and for manual deconfliction of pilot assignments. It has an "autofill" button to automatically generate pilot assignments, but there is a desire for more realistic and practically acceptable schedules. *Puckboard* accounts for pilot availability and general notions of required qualification levels. However, schedulers note that it is often more time-consuming to clean up the "autofill" output than to build a schedule manually. The work in this thesis will contribute to the automated scheduling functionality of *Puckboard*, which should increase the value-added to schedulers

Three general types of flights that schedulers need to assign pilots to are: simulation flights, local training flights, and missions. Simulation flights ("sims") are conducted with ground-based simulators. Sims and local training flights are primarily for pilots to complete training requirements. Missions generally last longer and can be given to schedulers with little advance notice. Schedulers manage several subcategories of flights, including, but not limited to, air-drop and air-refueling flights. For each flight, schedulers are given the scheduled times, locations, and required qualification levels. Schedulers do not have much leeway in adjusting flight times, so they are primarily concerned with a *crew assignment problem.*

Each flight has a set of minimum required qualification levels that must be satisfied by onboard personnel for the flight to be "flyable". Puckboard currently incorporates three broad categories of qualification levels: loadmasters, pilots, and aircraft commanders, but there are typically more than 30 unique qualification levels in a squadron. In addition to ensuring that all flights are adequately staffed, pilots have a syllabus of training requirements that need to be satisfied. Schedulers thus seek to ensure that as many pilot assignments as possible can contribute toward completing training requirements.

1.3 Prior Works

Although research focused solely on military crew scheduling is scarce, there are three relevant theses from the US Naval Postgraduate School (NPS). These works are most directly related to this thesis. In [6], the author constructs an integer program for joint flight scheduling and crew scheduling. They generate crew assignments that maximize the total "reward" across all pilots, where the reward is linked to completing training requirements. This work includes three general qualification levels (instructors, leads, and students), but *Puckboard* handles upwards of thirty qualification levels with more specialized qualifications, such as night or air-drop qualifications. In addition, this work does not explicitly consider crew rest, night flights, or robustness. Another thesis from the NPS maximizes the number of scheduled flights within the Fallon Range Training Complex, while considering airspace constraints, but does not provide detailed departure and arrival times [9]. The third thesis from the NPS augments [9] with detailed times and introduces the idea of *persistence*, wherein as disruptions occur, changes to previous schedules are minimized [13]. The idea of *persistence* is particularly relevant for Chapter 3.

In contrast to military crew scheduling, airline crew scheduling has received considerable attention from industry and academia. Airline crew scheduling refers to assigning pilots and flight attendants to flights. Typically, crew scheduling is the last step of the overall airline schedule planning process (schedule generation, fleet assignment, maintenance routing, crew scheduling) [2]. Recent work has looked at integrated flight and crew scheduling, including shifting flights five minutes earlier/later when creating crew schedules [10] and integrating aircraft routing and crew scheduling in an iterative process [15]. However, the current practice in industry is for crew scheduling to be performed after flight schedules have been fixed. This is also the case with Air Force crew scheduling, as flight schedules are fixed and determined by another department.

There are several other similarities between airline crew scheduling and military scheduling. Both have to consider rest requirements, leave/vacation, pilot qualifications, and pilot training. With military crew scheduling, the latter two are perhaps more involved, as there can be hundreds of different pilot qualifications and complex training syllabi. The way airline crew schedules are structured is significantly different from the military. (In the following discussion, crews refer to pilots or flight attendants.) Sequences of flights operated in the same day are grouped into *duty periods*. Then, since duty periods can start and end in different locations, sequences of duty periods are grouped into *pairings* [14]. Pairings typically last from one to five days and start and end at the same crew base. Notably, crew members typically stay together for the duration of a pairing. This is possible because the flights within each pairing require similar type ratings (e.g., Boeing 777/787 rating). The notions of duty periods and pairing are not as relevant in the military for a few reasons. First, the flight type, not just the aircraft type, dictates what pilot qualifications are required and consequentially which pilots are eligible to be assigned to it. For example, flights involving in-flight refueling require pilots with special qualifications. In addition, military pilots have detailed training requirements involving sequences of flights (with certain characteristics) that must be completed. Airline pilot training requirements are more general, like minimum flight hour requirements, making it easier to group crew members together.

Another distinction is that airline crew scheduling is usually divided into two subproblems to reduce computational complexity [1]. The crew pairing problem (CPP) combines flights and duties into valid crew pairings, taking into account duty constraints, and the crew rostering problem (CRP) assigns specific crew to these pairings.

The CPP can be solved before the CRP because only high-level metrics of the crew roster (e.g., number of crew at each crew base) are necessary to generate the pairings. In the military, this process would not work as each pilot has unique qualifications and training requirements that would need to be incorporated when solving the CPP. This necessitates the consideration of each pilot individually. For example, given two pilots A and B with the same qualification level, a flight may be useful for A because it completes an outstanding training requirement, but useless for B because they have already completed that requirement. Moreover, it is difficult to take crew preferences into account in the CPP since it is undecided who will be assigned to each pairing. As such, in this thesis, we do not follow the sequential crew pairing then crew rostering process.

One major feature of crew schedules is connection times between successive events operated by the same crew. Airlines have minimum connection times dictated by the Federal Aviation Administration (FAA) and labor regulations, while the military has similar regulations. In addition, there is a trade-off between efficiency and robustness when setting connection times. With shorter connection times, the schedule is more efficient since pilot and aircraft utilization are higher, but the schedule is less robust as delays are more likely to propagate. For example, if a flight (f') is delayed and the next flight (f) that the crew is scheduled to operate is scheduled soon after, f may also be delayed. In [15], the authors introduce a non-robustness metric, which penalizes crew connections times close to the minimum connection times. Other notions of robustness include buffer times and expected delay propagation [16], and move-up crews [12], which we will discuss in detail in Chapter 4.

More broadly, several prior works consider variations of scheduling problems. The classical problem of job-shop scheduling involves assigning machines of varying capability to tasks of varying length with the objective of minimizing the total makespan (length of the schedule) [8]. Scheduling problems are also studied in other disciplines. In healthcare, there is the doctor/nurse scheduling problem [17], where rest requirements and qualifications are important. In the patient appointment scheduling problem, just as pilots have preferences and a required set of events, patients prefer certain times and need to be matched with certain types of medical workers across a set of required appointments [5]. Railroad agencies also generate schedules and actively seek to mitigate disruptions. In [3], the authors formulate and solve a recovery-robust optimization problem for train timetabling, which is better suited to handle disruptions. In the trucking industry, collaborative scheduling across different companies and terminals has been explored [11]. In addition, public transit agencies undertake a multi-step scheduling process similar to airlines [4]. There are numerous examples of scheduling problems across other industries as well, but the main point is that military crew scheduling is relatively unexplored but can draw inspiration from several prior works, particularly in commercial aviation.

1.4 Outline

In Chapter 2, we present the baseline integer program that includes the metrics of overqualification and training requirements completed. In Chapter 3, we modify the baseline formulation to handle disruptions while preserving the original schedule as much as possible. We also discuss schedule visualizations and initial user feedback on our scheduling formulations. In Chapter 4, we present formulations that incorporate buffer time and move-up crews with the goal of creating more robust schedules. We end with conclusions and future work in Chapter 5.

Chapter 2

Baseline Integer Program

Air Force schedulers are given a slate of flights that must be filled by pilots. Each flight has a scheduled departure and arrival time, as well as pre-mission briefing and post-mission rest times. These scheduled times are typically determined beforehand based on aircraft availability and rotation, so there is little flexibility in adjusting these times. As such, the baseline integer program that we develop centers on crew assignment. Each flight has a minimum and maximum number of pilots that can be assigned to it based on aircraft size. In addition, each flight has a list of qualification requirements that must be satisfied. These qualification requirements can be as simple as “two pilots and three loadmasters” or as complex as “one A-level instructor pilot with special airdrop qualifications, two C-level pilots with right seat airdrop qualifications”. In addition, schedulers must account for pilot leave and temporary duty travel (TDY). Schedulers have various priorities when creating pilot assignments, including, but not limited to, completing pilot training requirements and assigning the lowest qualified pilots to flights.

2.1 Notation

We define the following sets and subsets. We have sets for pilots, flights, time periods, and qualification levels, and various subsets based on them. We choose hours as the units for the time periods, but other time intervals can easily be used instead.

$i \in I$	Set of Pilots
$f \in F$	Set of Flights
$t \in T$	Set of Time Periods
$q \in Q$	Set of Qualification Levels
$F_i \subset F$	Flights that pilot i is qualified for
$T_f \subset T$	Time periods in flight f
$U_f \subset F$	Flights that overlap with flight f
$I_f \subset I$	Pilots that are qualified for an assignment on flight f
$Q_f \subset Q$	Qualification level requirements of flight f
$I^{EP} \subset I$	Pilots qualified to be evaluator pilots
$I^{IP} \subset I$	Pilots qualified to be instructor pilots

We also define the following parameters for flight requirements, pilot availability, and pilot qualifications.

I_f^{min}	Minimum number of pilots required for flight f
I_f^{max}	Maximum number of pilots allowed for flight f
a_{it}	1 if pilot i is available during period t; 0 otherwise
z_{iq}	1 if pilot i satisfies qualification q; 0 otherwise
N_{fq}	Number of pilots with qualification q or higher needed on flight f

2.2 Baseline Formulation

We denote the binary decision variable as X_{if} which is 1 if pilot $i \in I$ is assigned to flight $f \in F$, and 0 if pilot i is not assigned to f. We start with the baseline formulation where the objective function is to minimize the number of pilot assignments.

$$\min \quad \sum_{i \in I} \sum_{f \in F_i} X_{if} \tag{2.1}$$

subject to:

$$\sum_{i \in I_f} X_{if} \geq I_f^{min} \qquad \forall f \in F \tag{2.2}$$

$$\sum_{i \in I_f} X_{if} \leq I_f^{max} \qquad \forall f \in F \tag{2.3}$$

$$\sum_{i \in I_f} z_{iq} X_{if} \geq N_{fq} \qquad \forall f \in F, q \in Q \tag{2.4}$$

$$X_{if} + X_{i\overline{f}} \leq 1 \qquad \forall i \in I, f \in F, \overline{f} \in U_f \tag{2.5}$$

$$X_{if} \leq a_{it} \qquad \forall i \in I, f \in F, t \in T_f \tag{2.6}$$

$$X_{if} \in \{0, 1\} \qquad \forall i \in I, f \in F \tag{2.7}$$

Constraints 2.2 and 2.3 ensure that the number of pilots assigned to each flight is within the acceptable range. Constraint 2.4 ensures that each flight has enough pilots with the appropriate qualification levels. Consider a flight that needs one each of qualification levels (A,B,C) with A being the highest qualification level and C being the lowest. Then, $N_{fA} = 1$, $N_{fB} = 2$, $N_{fC} = 3$. We require three pilots to have qualification C or higher because if we required only one, then the pilot with qualification level A or higher would satisfy this requirement. That is, if $N_{fA} = 1$, $N_{fB} = 1$, $N_{fC} = 1$, one pilot with qualification level A would satisfy all of these constraints. Constraint 2.5 ensures that pilots are not assigned to overlapping flights, while constraint 2.6 ensures that pilots are available (i.e., not on leave or TDY) during flight assignments. Finally, constraint 2.7 sets X as a binary variable.

2.3 Overqualification

We now discuss two possible modifications to the objective function. Schedulers may prefer to minimize the utilization of their highest qualified pilots to preserve schedule flexibility, as higher-qualified pilots are eligible to be assigned to more flights. In addition, lower-qualified pilots need more training and are often more eager to fly. One simple way to encode this preference is to modify the objective function as follows. Equation 2.8 minimizes the utilization of the two highest qualified classes of pilots: instructor and evaluator pilots.

$$\min \quad \sum_{i \in I^{IP}} \sum_{f \in F} X_{if} + \sum_{i \in I^{EP}} \sum_{f \in F} X_{if} \tag{2.8}$$

Schedulers may also want to use the lowest qualified feasible pilot for each assignment. We define an overqualification penalty $O_{if} \in [-1, 0)$ for each feasible pilot i and flight f combination, as shown below.

$$O_{if} = -\frac{\text{rank}_{if}}{N_f} \tag{2.9}$$

We define rank_{if} as the rank of pilot i's qualification level among all acceptable qualification levels for flight f (with higher ranks corresponding to higher-qualified pilots). N_f is the number of feasible qualification levels for flight f. Thus, the highest qualified pilot for a flight with three eligible qualification levels will have $O_{if} = -1$, and the lowest qualified pilot will have $O_{if} = -1/3$. We switch to a maximization objective function (to be consistent with subsequent objective functions). Equation 2.10 maximizes the sum-product of X_{if} and O_{if}. Since this product is negative, it minimizes the overqualification across all assignments. Thus, whenever choosing between two feasible pilots with different qualification levels, the model will prefer the pilot with lower qualification level, or equivalently, a less negative O_{if}.

$$\max \quad \sum_{i \in I} \sum_{f \in F_i} X_{if} O_{if} \tag{2.10}$$

We note that the specification of O_{if} is amenable to different preferences. For example, users may prefer overqualification values of different but very close qualification levels to be the same. In addition, users may prefer overqualification values to depend on the type of flight being assigned, in addition to the qualification level of the pilot.

2.4 Training Requirements

While some flights fulfill specific Air Force missions such as cargo flights, other flights are for the purpose of completing training requirements. Each pilot has a set of training requirements that they must complete within a certain time period or on a rolling basis. Oftentimes, a pilot needs to complete a requirement multiple times in a given time period. Note that we distinguish training requirements from qualification requirements, as the latter describes the required qualification levels to adequately crew a flight. We define the following sets and parameters to formulate training requirements.

S	Set of types of training requirements
$F_s \subset F$	Flights that complete training requirement $s \in S$
$S_i \subset S$	Set of training requirements that pilot $i \in I$ needs to satisfy
R_{is}	Number of requirements of type $s \in S$ that pilot $i \in I$ needs
T_{is}	Time until requirements of type $s \in S$ are due for pilot $i \in I$

We define an additional binary variable r_{kis} which is 1 if k number of requirements of type s are satisfied by pilot i, and 0 otherwise. The objective function for maximizing training requirement completion is as follows.

$$\max \quad \sum_{i \in I} \sum_{s \in S_i} \sum_{k \in 1:R_{is}} r_{kis} \tag{2.11}$$

We also introduce additional constraints to define r_{kis}. The constraints in the baseline formulation still apply.

$$\sum_{k=1:R_{is}} r_{kis} <= \sum_{f \in F_s} X_{if} \qquad \forall i \in I, s \in S_i \tag{2.12}$$

$$r_{kis} <= r_{k-1,i,s} \qquad \forall i \in I, s \in S_i, k \in 2 : R_{is} \tag{2.13}$$

$$r_{kis} \in \{0, 1\} \qquad \forall i \in I, s \in S_i, k \in R_{is} \tag{2.14}$$

Constraint 2.12 states that the number of type s requirements completed (summation of r for a given pilot i and requirement type s) is less than or equal to the number of flights flown of type s by pilot i. Constraint 2.13 defines the order in which r variables are set to 1: the binary variable representing one requirement satisfied of type s (r_{1is}) must equal 1 before the binary variable representing two requirements satisfied (r_{2is}) can be set to 1. That is, if k requirements of type s are completed by pilot i, then $k-1$ requirements of type s have also been completed. Finally, constraint 2.14 defines r_{kis} to be a binary variable.

Besides summing the number of requirements completed, there are several other possible objective functions. Two are listed below.

$$\max \quad \sum_{i \in I} \sum_{s \in S_i} \sum_{k \in 1:R_{is}} r_{kis}(R_{is} - k + 1) \tag{2.15}$$

$$\max \quad \sum_{i \in I} \sum_{s \in S_i} \sum_{k \in 1:R_{is}} r_{kis}(R_{is} - k + 1)(\frac{1}{T_{is}}) \tag{2.16}$$

Equation 2.15 does not reward each training requirement completed equally, whereas equation 2.11 does. Instead, requirements completed have a higher reward if there are more outstanding training requirements. For example, if pilot i has three outstanding requirements of type s ($R_{is} = 3$), the first requirement completed will be rewarded three ($R_{is} - k + 1 = 3$, since $k = 1$); the second requirement completed will be rewarded two ($R_{is} - k + 1 = 2$, since $k = 2$); and the third will be rewarded one. This prioritizes training requirements where pilots have several training requirements of a particularly type outstanding. In the Air Force, training requirements can often be waived if the number of outstanding requirements is sufficiently low, so it is important to reduce the higher number of outstanding requirements as quickly as possible. Equation 2.16 is similar to equation 2.15 but includes T_{is} which represents the time remaining until requirement s is due for pilot i. Thus, requirements due sooner will be prioritized over requirements due later.

2.5 Results

We tested our formulation on a sample, anonymized dataset. The dataset contained 87 pilots with 31 possible qualification levels. We also obtained a sample training requirements schedule, which included a) semi-annual requirements to be completed within the first or second-half of the calendar year and b) rolling requirements to be completed, for example, every three months. The dataset also contained details on flight types and minimum required qualifications levels of onboard personnel. We also aggregated leave and TDY data to determine pilot availability.

Two main objectives we have discussed so far are minimizing overqualification and maximizing training requirements. We expect that there is a trade-off between overqualification and training requirements. Minimizing overqualification amounts to assigning the lowest-qualified feasible pilot for each assignment. As such, the lower-qualified pilots will be heavily utilized, while the higher-qualified pilots will be kept in reserve. Since there are a finite number of training requirements for each pilot, repeatedly scheduling the lower-qualified pilots may lead to fewer training requirements completed than if the schedule was more evenly distributed across qualification level. Koch's thesis provides a thorough breakdown of the trade-offs between overqualification and training requirements [7]. Key results from joint work are shown below, but the main results of this thesis are in Chapter 3 and 4 with disruption handling and robustness.

We show results for 10 weeks wherein 1,134 pilot assignments were made. The schedule is determined one week at a time, so the optimizer is run 10 times. In each week, we schedule all of the flights that depart within that week. Note that some longer flights will spillover into subsequent weeks. Thus, when determining pilot availability, we account for pilot assignments that spill over from one week to the next. We tested three objective functions: overqualification (2.10), training requirements with a linear penalty (2.15), and an objective that maximizes the sum of overqualification and training requirements (2.10 + 2.15). Recall that overqualification was defined as a maximization objective in 2.10 because O_{if} is negative. Figure 2-1 shows the total overqualification each week for the three objectives. We do not use overqualification score as in 2.9, but instead show overqualification as the $rank_{if}$ of the assigned pilot i minus the required qualification level for flight f that pilot i is satisfying. The trends in the plot would be similar with overqualification shown either way, but this difference method should be more intuitive. Note that because of constraint 2.4, this difference will always be non-negative. Figure 2-2 is formatted similarly to Figure 2-1, but shows the number of training requirements completed each week for the three objectives.

Figure 2-1: Total overqualification across 10 weeks with three different objectives.

We first note that the overqualification values remain relatively similar across weeks for each objective. However, the number of training requirements completed decreases as time increases, since there are fewer unsatisfied training requirements in later weeks. We see that the training requirements objective has the highest overqualification and highest number of training requirements completed. On the other hand, the overqualification objective has the lowest overqualification and the lowest number of training requirements completed. The objective that includes both of them is in between these two objectives, for both overqualification and training requirements completed. These results display the aforementioned trade-off between overqualification and training requirements. With different weightings, the space between the overqualification and training requirements objective can be spanned.

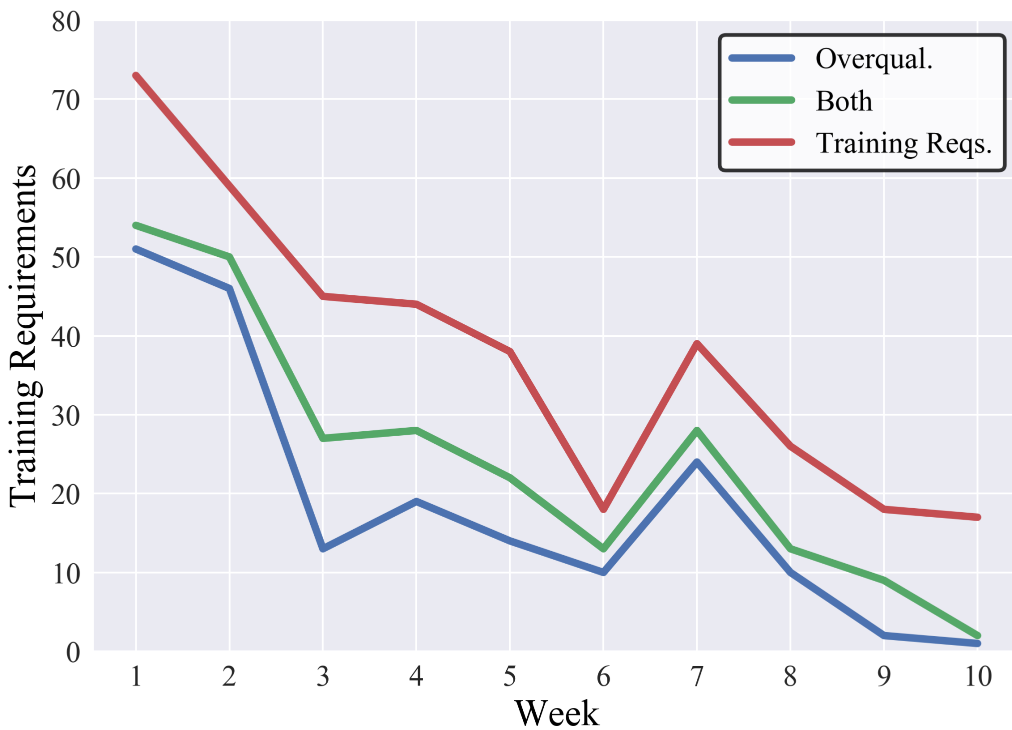

Figure 2-2: Number of training requirements completed across 10 weeks with three different objectives.

As minimizing overqualification and completing training requirements are both important, we expect that the most useful objective is a combination of the two. The precise weighting between the two is a user choice, and Koch explores this issue further. Optimizing for only overqualification could be useful if a scheduler, for example, knows that several training requirements can be waived. On the other hand, if a scheduler is not expecting many pop-up flights, they may decide to optimize for training requirements only since keeping higher-qualified pilots available is not as important.

Chapter 3

Disruption Handling

The baseline formulation in Chapter 2 provided a methodology for "clean-slate" scheduling. Oftentimes, though, disruptions occur that make a previously planned schedule infeasible. In this chapter, we discuss a formulation for handling disruptions with minimal change to the original schedule. The constraints of the baseline integer program still apply. We then discuss takeaways from the user studies we conducted.

3.1 Disruption Formulation

We define the following notation. We define the "original schedule" as the pre-disruption schedule and the "new schedule" as the post-disruption schedule.

D	set containing pairs (i, f) where pilot i is unavailable for flight f
V	set containing pairs (i, t) where pilot i is unavailable during t
X_{if}	1 if pilot i is originally assigned to flight f
Y_{if}	1 if pilot i is assigned to flight f in new schedule
Z_{if}	1 if pilot i assignment to flight f changed in new schedule
O^*	objective value of original schedule
l	tolerated decrease in objective value with new schedule

A pilot could be unavailable for a specific set of flights (D) because of the location of the flights, or a pilot could be unavailable for all flights within a specific set of time periods (V). The variable Z tracks changes between the original schedule (X) and the new schedule that will be generated (Y). The objective is to minimize the number of changes between the old and new schedule. This is because schedule changes put a burden on both the scheduler, who has to communicate these changes, and the pilots, who may go from being off-duty to on-duty, or vice versa. The objective can be represented as follows, with terms for overqualification and training requirements with respective weights of c_o and c_r.

$$\max \; -\sum_{i \in I} \sum_{f \in F} (Z_{if} - c_o X_{if} O_{if}) + \sum_{i \in I} \sum_{s \in S_i} \sum_{k \in 1:R_{is}} c_r r_{kis} \tag{3.1}$$

The following constraints must hold.

$$Z_{if} \geq Y_{if} - X_{if} \quad \forall i \in I, f \in F \tag{3.2}$$

$$Z_{if} \geq -(Y_{if} - X_{if}) \quad \forall i \in I, f \in F \tag{3.3}$$

$$\sum_{i \in I} \sum_{f \in F_i} c_o X_{if} O_i + \sum_{i \in I} \sum_{s \in S_i} \sum_{k \in 1:R_{is}} c_r r_{kis} \geq O^* - l \tag{3.4}$$

$$Y_{if} = 0 \quad \forall (i, f) \in D \tag{3.5}$$

$$a_{it} = 0 \quad \forall (i, t) \in V \tag{3.6}$$

Constraints 3.2 and 3.3 define Z_{if} to be 1 whenever X_{if} and Y_{if} differ, indicating a new assignment for pilot i on flight f ($Y_{if} = 1, X_{if} = 0$) or a cancelled assignment ($Y_{if} = 0, X_{if} = 1$) in the new schedule. We weight new assignments and cancelled assignments the same, but this could be adjusted. Constraint 3.4 ensures that the objective value does not decrease more than tolerance l in the new schedule, relative to the original schedule. Lastly, constraints 3.5 and 3.6 enforce the pilot unavailability constraints imposed by the disruptions, specifying flights and time periods, respectively, that pilots cannot be assigned to.

3.2 User Studies

To evaluate whether our formulations up to this point made sense, we conducted as set of user interviews with Air Force schedulers. We presented schedulers with several disruption scenarios wherein pilot availability was reduced. We then gave schedulers several new schedule options with different weightings between number of changes, overqualification, and training requirements.

3.2.1 Schedule Visualization

The first step was to design a schedule visualization tool to present to schedulers. We did not use the existing Puckboard user interface to display our schedules because we wanted to focus on our scheduling algorithms rather than a specific user interface. We also wanted to create a customized visualization that emphasized metrics of interest, like overqualification and training requirements. Figure 3-1 shows an overview of the Gantt Chart visualization that we presented in our user studies. Pilots were listed in

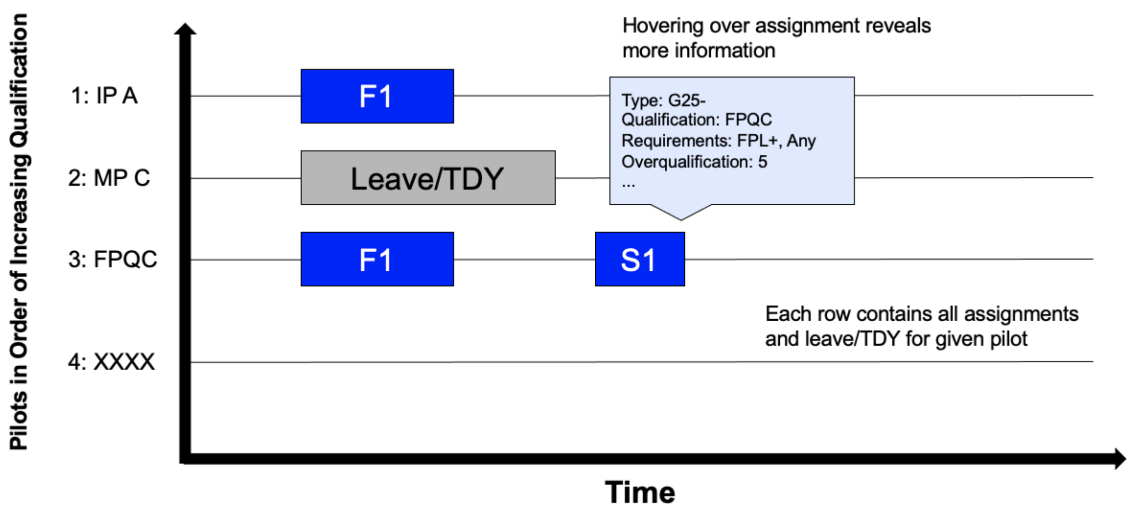

Figure 3-1: Gantt Chart layout, as presented during user studies.

increasing order of qualification level on the y-axis (with the highest qualified pilots on top). Pilot names were anonymized and their qualification levels were provided. Flight events (denoted by "F") and simulation events ("S") are shown for each pilot, including leave/TDY assignments. We presented these plots using interactive HTML files using *plotly* in Python. Hovering over pilot assignments revealed more information, like the type of flight, required qualification levels, and the overqualification score for an assignment. For the visualization, we used a simpler overqualification score, which was simply the difference between pilot qualification level and required qualification level of the role they were assigned, as in Figure 2-1.

Figure 3-2 shows an example of a Gantt Chart schedule option. The schedule shows 1) the disruptions in pilot unavailability (red lines), 2) pilot leave/TDY (grey), 3) pilot assignments that remained the same between the old and new schedule (blue), 4) pilot assignments that were present in the original schedule but not in the new schedule (red), and 5) pilot assignments that were not present in the original schedule but present in the new schedule (green). In this example, three pilots become

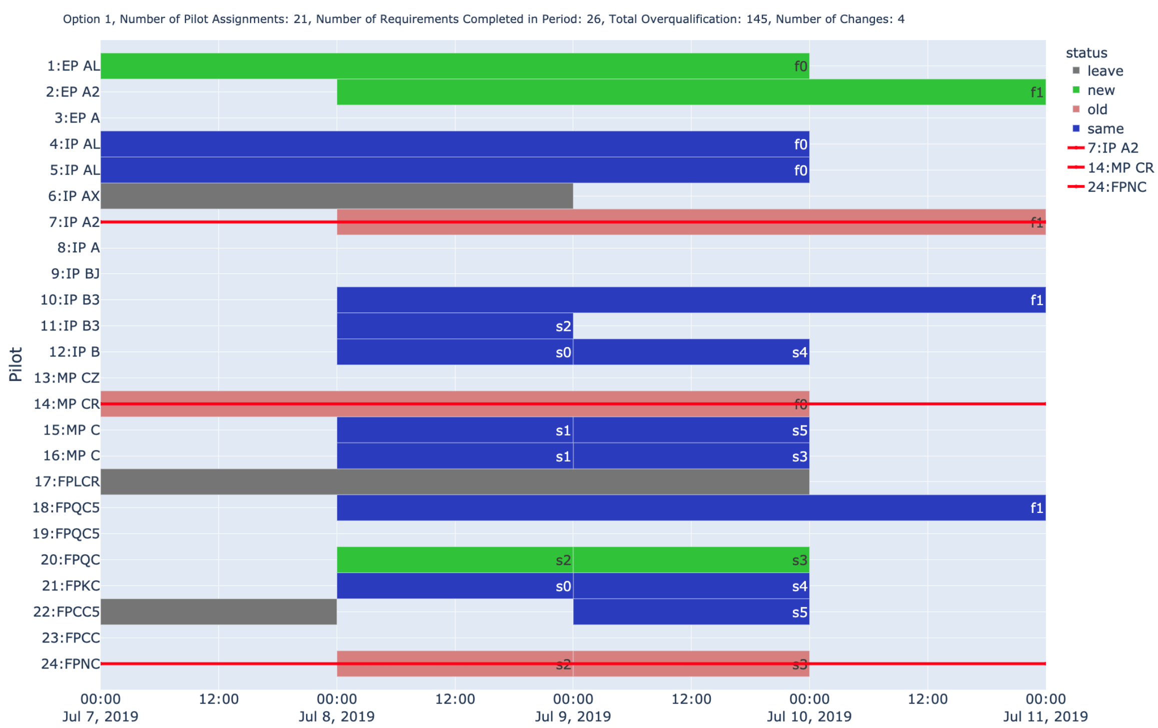

Figure 3-2: Example of Gantt Chart with three pilot disruptions.

unavailable for the duration of the schedule window due to disruption, so they are replaced by available pilots. For example, pilot "24:FPNC" is replaced by "20:FPQC" on simulation flights 2 and 3. We also provided schedule metrics, including the number of changes, overqualification, and number of requirements completed. Finally, we included a change log that listed all of the differences between the old and new schedule (not shown).

3.2.2 Preliminary User Feedback

Schedulers validated our metrics of number of changes, overqualification, and number of requirements, and said they were useful. We list recurring items of feedback that we heard, along with accompanying model augmentations to address their feedback. While we have formulations to address most of the user feedback, explicit implementation of these is not shown in this thesis.

Reassignment Cost: Schedulers indicated that the time remaining until an assignment was set to occur impacted their willingness to change an assignment. Specifically, schedulers were more reluctant to change assignments that were occurring sooner, as tracking down pilots and confirming the change can be a hassle with short notice. Schedulers were also more willing to change simulation assignments rather than mission assignments, which are considered more important. Schedulers were also more reluctant to change the schedule of a higher-qualified, senior pilot than a lower-qualified pilot. In addition, they preferred to modify assignments of shorter flights before those of longer flights. To incorporate these factors, we can have a reassignment cost attached to each pilot assignment in the original schedule that is a function of time remaining, qualification of currently assigned pilot, duration of flight, and flight type.

Training Requirements: Schedulers indicated that they prioritize training requirements completion for higher-qualified pilots, as there is a trickle-down effect wherein re-qualifying higher-qualified pilots sooner can enable them to train lower-qualified pilots earlier. To account for this, we can further modify the requirement completion reward (equation 2.16) to include the qualification level of the pilot. In addition, while schedulers have little control of flight times, they can control the type of *simulator flight* that is conducted. Different simulator types complete different training requirements, so altering the simulator type can be useful for expediting certain training requirements. To model this, we can include an additional decision variable for each sim to modify the simulator type.

Overqualification: Schedulers appreciated overqualification as a metric to ensure efficient assignments, and as a way to preserve higher-qualified pilots. They also noted that they want to preserve scarce pilot groups (regardless of qualification level) so that they have a diverse set of pilots off-duty at any point in time. We can introduce a utilization metric for each pilot group and minimize the maximum utilization in the objective. However, note that with a maximization objective, this turns into max-min where the utilization metrics are negative and higher utilizations are more negative.

Relative Importance: Schedulers generally preferred the solutions that came from objectives which incorporated all three terms: number of changes, overqualification, and training requirements. They emphasized that the relative importance of these metrics is situation- and squadron-dependent. Naturally, if a squadron is behind in training, completing training requirements will take precedence, whereas if a squadron is expecting lots of unexpected pop-up missions, overqualification will be most important. For the *Puckboard* implementation, they suggested a feature for schedulers to input their desired weightings. In addition, they recommended that we develop functionality for the optimization model to infer/suggest appropriate weightings between the metrics.

Chapter 4

Robustness

In Air Force crew scheduling, several degrees of uncertainty may disrupt the schedule. In Chapter 3, we introduced a disruption handling formulation. This handles disruptions as they arise. However, there is also a desire to develop schedules from the start that are more robust to possible disruptions. In this chapter, we present two notions of robustness, and formulations that incorporate them.

4.1 Degrees of Uncertainty

Schedulers contend with several possible underlying uncertainties. Below is a non-exhaustive list of a few types of uncertainties. We comment on the similarities and differences in the way these uncertainties manifest in commercial aviation and Air Force scheduling.

1. **Flight Duration:** Each flight has a planned duration, defined as the difference between the scheduled arrival and scheduled departure times. In commercial aviation, the actual flight duration may exceed the planned flight duration due to various factors like weather, crew/passenger connection delays, or maintenance. In the Air Force, the actual flight duration can be extended for the same reasons, but also for additional ones. For example, a VIP escort mission may be extended by a day for operational reasons. In addition, training flights may be extended if the intended training has not been completed yet due to,

for example, weather conditions. These factors make the flight duration of Air Force flights more uncertain than in commercial aviation.

2. **Pilot Availability:** The availability of pilots may also be uncertain. Personal leave due to emergencies can make pilots unavailable at short notice. In addition, schedulers may not know the true pilot availability of their roster. In commercial aviation, this is less of an issue, as the crew scheduling department has direct insight into pilot availability, and large control over it. However, with Air Force crew scheduling, schedulers are aggregating pilot availability across different data sources and are less certain about true pilot availability. For example, pilots have non-flying duties which schedulers may not be aware of that could take precedence over a training flight. For these reasons, schedulers prefer schedules where back-up pilots are available to cover for unavailable pilots.

3. **Pop-up Flights:** As the Air Force operates in a dynamic environment, last-minute flights ("pop-up flights") may need to be crewed. In commercial aviation, this is far less common as flight schedules are published well in advance because of the need to sell tickets to passengers. Schedulers thus have to be mindful of the possibility that they will need to crew additional flights.

4.2 Buffers and Delay Propagation

We borrow two notions from commercial aviation, buffer and delay propagation, to increase robustness to flight duration uncertainty. Pilots are often assigned to consecutive flights with a minimum rest requirement between them. If a flight is delayed, subsequent flights operated by the same crew or same aircraft may also be delayed. Buffer time refers to the ability to absorb flight delays such that subsequent flights are not delayed, and delay propagation refers to when delay is propagated to subsequent flights. Consider a case where a pilot is scheduled on flight f' followed by flight f. If f' is delayed due to a longer than planned flight duration, f may be delayed while it waits for this pilot to complete f' and satisfy minimum rest requirements. Alternatively, this crew member may need to be replaced, representing a schedule disruption. The goal here is to develop schedules that are robust to flying time variability. We define the following notation to formally describe the scenario.

d_f	scheduled departure time of f
a_f	scheduled arrival time of f
μ_f	mean duration of f
e_f	actual duration of f
MINSIT_f	minimum rest time after flight f
T_{buffer}	threshold below which buffer time is counted
$\text{BufferTime}_{iff'}$	buffer time between flight f' that precedes flight f for pilot i
$h_{iff'}$	expected delay propagation from f' to f for pilot i

4.2.1 Buffers

To account for variability in flying time, we define μ_f as the mean duration of flight f. The scheduled times d_f and a_f are determined beforehand. In commercial aviation, airlines oftentimes set a_f such that $a_f > d_f + \mu_f$ to build "slack" into the schedule. Since flight duration may be more uncertain in the Air Force, it is expected that $d_f + \mu_f > a_f$ is more common, meaning that the expected flight duration exceeds the scheduled flight duration. Fig. 4-1 shows an example where a pilot is assigned to flights f' and f where f' precedes f. In this example, $a'_f < d_{f'} + \mu_{f'}$. However, note that the notion of buffer time is still relevant even if this condition does not hold.

The expected arrival time plus the minimum sit time ($d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}$) yields the earliest time that a pilot assigned to f' is ready for f. We define the quantity below as the *buffer time* between successive flights f' and f operated by pilot i. Note that we include i in the subscript to indicate that pilot i is assigned to both f' and f.

$$\text{BufferTime}_{iff'} = d_f - (d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}) \tag{4.1}$$

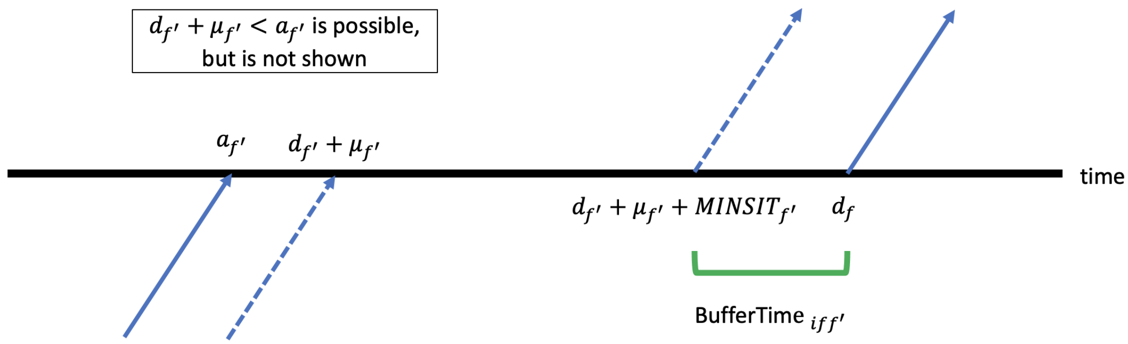

Figure 4-1: Example of buffer between flights f' and f.

The buffer time is equal to the maximum amount that e_f (the actual flying time of f') can exceed $\mu_{f'}$ such that flight f will not be delayed because of the crew coming from f'. To minimize the chance of disruptions, we would like to assign pilots to pairings with high buffer times. Flight pairings with a higher buffer are less likely to have delay propagate due to late arriving flights. On the other hand, with higher buffer times, pilot utilization will decrease, as they will have more idle time. The set P_{if} contains a list of flights that precede f for which buffer time is relevant for pilot i. Three conditions must be satisfied for a flight f' to be part of P_{if}.

1. Pilot i meets at least one qualification requirement for both f' and f. We do not create buffer variables for a pilot that is not qualified for both f' and f.

2. $d_f \geq d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}$. Flight f departs at the same time or after the time that pilot i is ready for f after flying on f'.

3. $\text{BufferTime}_{ff'} \leq T_{\text{buffer}}$. The buffer time between flights f' and f is less than or equal to T_{buffer}, the buffer threshold. We choose 5 hours as the threshold below which we count buffer times, but this can be adjusted as desired. With a larger threshold, computational complexity increases, but buffer time is tracked across more flight pairings. The choice of T_{buffer} depends on the magnitude of delays expected.

When pilots are assigned to both flight f' and flight f in P_{if}, we want to penalize assignments where f' and f are "tight". Recall that we have a maximization objective function. Thus, as buffer time decreases, the corresponding term in the objective function should get more negative. For flight pairings where the buffer time exceeds 5 hours (the threshold that we chose), we do not apply any penalty in the objective. This means that we are indifferent between having a buffer of 5 hours and 6 hours, because while a larger buffer time is more robust to disruptions, it also means more pilot downtime. More complex, non-monotonic buffer penalties and thresholds could of course be incorporated. We define $b_{iff'}\ \forall f \in F, f' \in P_{if}$ as the buffer penalty for assignments of pilot i to flight f' followed by flight f. Note that $b_{iff'}$ is the *buffer penalty*, which is dependent but not equal to buffer time, whereas $\text{BufferTime}_{iff'}$ is equal to the *buffer time* between f' and f.

$$b_{iff'} = -\frac{T_{\text{buffer}} + 1 - \text{BufferTime}_{iff'}}{T_{\text{buffer}} + 1} \tag{4.2}$$

When $\text{BufferTime}_{iff'} = 0$, $b_{iff'} = -1$, and when $\text{BufferTime}_{iff'} = T_{\text{buffer}}$, $b_{iff'} = -\frac{1}{T_{\text{buffer}}+1}$. Thus, shorter buffer times have more negative penalties. We define a binary indicator variable $B_{iff'}$ which is 1 if pilot i is assigned to both flight f and f', where f' is the assignment for pilot i directly preceding f, and 0 otherwise. We define the objective term for buffer penalty (BP) as the summation across all pilots and their possible f' and f pairs, as shown below.

$$BP = \sum_{i \in I} \sum_{f \in F_i} \sum_{f' \in P_{if}} b_{iff'} B_{iff'} \tag{4.3}$$

In equation 4.3, parameter b represents the buffer penalty, whereas indicator variable B shows whether pilot i was assigned to f and f'. We now need to mathematically define B. We cannot directly encode B with X, like with 1) $B_{iff'} \leq X_{ifq}$ and 2) $B_{iff'} \leq X_{if'q}$. Since we have negative buffer penalties (b) and a maximization objective, the solver would always set $B = 0$. Instead, we apply the following Big-M constraints $\forall i \in I, f \in F_i, f' \in P_{if}$, where $M > 2$ and $\epsilon = 0.01$.

$$X_{if} + X_{if'} \geq 2 - M(1 - B_{iff'}) \tag{4.4}$$

$$2 - \epsilon \geq X_{if} + X_{if'} - M(B_{iff'}) \tag{4.5}$$

These constraints ensure that if and only if $X_{ifq} + X_{if'q} \geq 2$, $B_{iff'} = 1$; otherwise, $B_{iff'} = 0$. When pilot i is assigned to both f and f' where f' precedes f, buffer penalties are included in the objective. We also need to make sure that we only include buffers where flight f' is the assignment of pilot i that immediately precedes their assignment to flight f. For example, if pilot i is assigned in order to flights (e, f', f), we only want to count the buffer between (e and f') and (f' and f), not between (e and f). The following constraint ensures that if flight f' is scheduled to depart after flight e (i.e., it is feasible for pilot i to be assigned to e, f', f in that order), we count the buffer between f' and f, if applicable, before counting the buffer between e and f. If $B_{iff'} = 1$ because pilot i is assigned to f' and f, then $B_{ife} = 0$, as the buffer time between e and f is not relevant since f' is scheduled between them.

$$B_{ife} \leq 1 - B_{iff'} \quad \forall (e, f') \in P_{if} \mid d'_f > a_e \tag{4.6}$$

4.2.2 Delay Propagation

We record buffer times when the scheduled departure time of f is later than the earliest ready time of a pilot assigned to flight f' preceding f (i.e., $d_f \geq d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}$). However, it is possible for the scheduled departure time of f to be before the expected earliest ready time. In this case, delay will propagate from flight f' to f, assuming that $e_f = \mu_f$ and no crew swaps are made. Figure 4-2 shows an example of this.

We define the quantity of *expected delay propagation* below. Expected delay propagation can be thought of as "negative buffer".

$$\text{DelayProp}_{ff'} = (d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}) - d_f \tag{4.7}$$

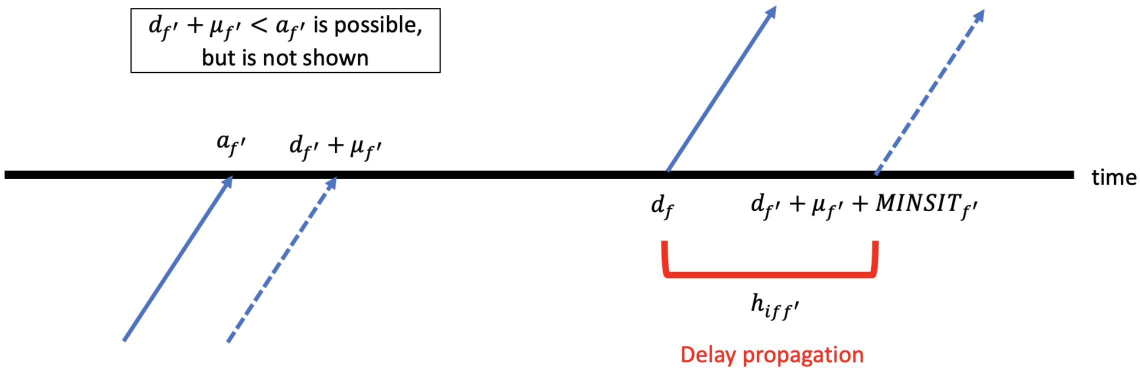

Figure 4-2: Example of expected delay propagation between flights f' and f.

Delay propagation is a common phenomenon in commercial aviation. Assigning pilots to sequences of flights where delay propagation is likely to occur based on mean flight durations is highly undesirable. In commercial crew scheduling, the crew pairing generation step would likely filter out pairs of assignments where there is expected delay propagation, so this explicit incorporation of delay propagation shown here may not be necessary. However, in Air Force crew scheduling, as aforementioned, flight duration is more variable and we expect that it may be desirable to directly encode delay propagation into the formulation.

The set Q_{if} is analogous to P_{if} and contains a list of flights that precede f for which delay propagation is possible. For each flight f' in Q_{if}, 1) pilot i is qualified for f', 2) $d_f < d_{f'} + \mu_{f'} + \text{MINSIT}_{f'}$ and 3) $0 \leq d_f - a_{f'} \leq T_{\text{DelayProp}}$, where $T_{\text{DelayProp}}$ is a user chosen threshold (we set it to 24 h). These are flights whose scheduled arrival times are sufficiently close to d_f such that delay could propagate to f and are expected to given μ_f.

We define $h_{iff'}$ $\forall i \in I, f \in F, f' \in Q_{if}$ as the penalty for expected propagated delay from flight f' to flight f. We define $h_{iff'} = -\text{DelayProp}_{ff'} - 1$, where the -1 constant is added to ensure that 1 hour of expected delay propagation has a more negative penalty ($h_{iff'} = -2$) than 0 hours of buffer ($b_{iff'} = -1$). $H_{iff'}$ is defined identically to $B_{iff'}$ except with Q_{if} rather than P_{if}. The delay propagation penalty

(DP) is shown below.

$$DP = \sum_{i \in I} \sum_{f \in F_i} \sum_{f' \in P_{if}} h_{iff'} H_{iff'} \tag{4.8}$$

4.3 Move-up Crews

The implicit assumption when counting delay propagation is that the original pilot assignments are kept in the face of disruptions. However, in commercial aviation, airlines often shuffle or swap pilot assignments. In this subsection, we describe the concept of move-up crews and how to incorporate them in the formulation. Figure 4-3 shows an example of a move-up crew. Suppose that pilot i (in blue) is assigned to flight f_1 then f_2 and pilot j (in orange) is assigned to g_1 then g_2. If flight f_1 is delayed, this delay could propagate to f_2 since d_{f_2} is infeasible, given the minimum sit time. Alternatively, it may be possible to *move-up* pilot j to flight f_2. Pilot j is called a move-up crew because they are moving to an earlier assignment. Note that utilizing a move-up crew is generally preferable to calling up an off-duty crew. Ideally, pilot i can cover g_2 and thus swap assignments with pilot j, but we do not require this. Besides providing potential swapping opportunities, a move-up crew allows for a choice of flights to cancel, if a cancellation is deemed necessary. For example, with pilot j as a move-up crew for flight f_2, either g_2 (if pilot j moves-up to f_2) or f_2 can be cancelled.

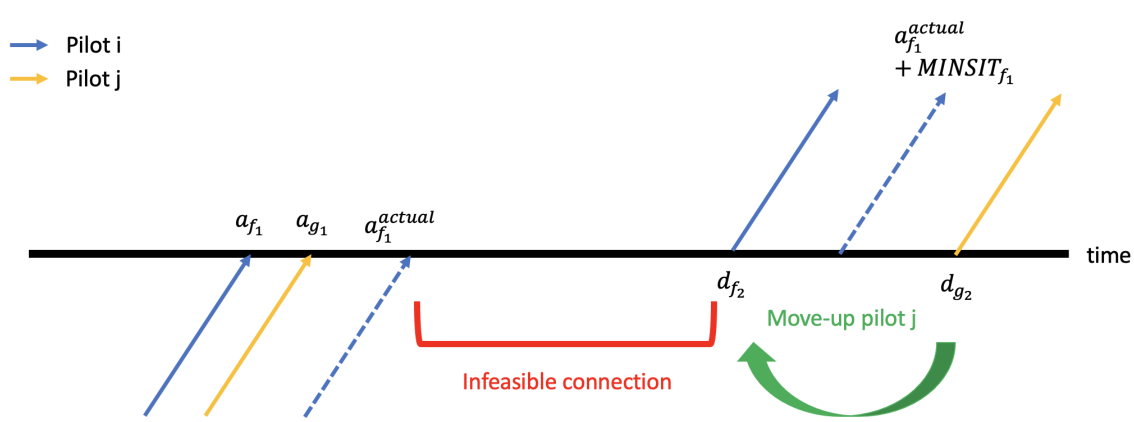

Figure 4-3: Example of option for pilot j to move-up to pilot i's assignment on f_2.

The goal is to count the number of move-up crew options for every pilot i assigned to f. When counting move-up crew options, we only count pilots with equal or greater qualification level as pilot i because our decision variable X does not track the qualification required fulfilled by each pilot on each flight. For example, suppose the qualification levels in decreasing order are (A, B, C), and pilot i has qualification level B. The move-up crew options for pilot i will have the same qualification level (B) or higher (A). This assumption neglects the fact that pilots with qualification level C may be able to substitute for pilot i on assignments where only qualification level C is required. We tested a formulation where the decision variable includes the qualification requirement being filled (i.e., $X_{ifq} = 1$ if pilot i fulfills qualification q on flight f), but it was not computationally tractable. In addition, because of the overqualification term in the objective, assigned pilots usually have qualification levels similar to the required qualification level.

Consider the scenario shown in Figure 4-4 where pilot i is assigned to flight f_1, f_2, f_3 in order and pilot j is assigned to g_1, g_2, g_3 in order. If f_1 is delayed, we may be able to use pilot j as a move-up option for flight f_2. To ensure that pilot j is a move-up option for pilot i on flight f_2, we need to ensure the following conditions are met:

1. Pilot j has qualification level equal to or higher than pilot i.

2. Pilot j is available for the duration of f_2 (provided that they vacate their assignment on g_2).

   (a) Pilot j is available at the start of f_2. This means that they are not on leave or assigned to any flights that start before f_2 and overlap with f_2.

   (b) Pilot j is available through the end of f_2. To satisfy this, we require that the end time of g_2 is greater than or equal to the end time of f_2. For a more comprehensive formulation without this assumption, see Section 4.3.1.

We assume that Condition 1 is satisfied in Figure 4-4, so pilot j has qualification level greater than or equal to pilot i. Condition 2a) is satisfied because g_1 does not

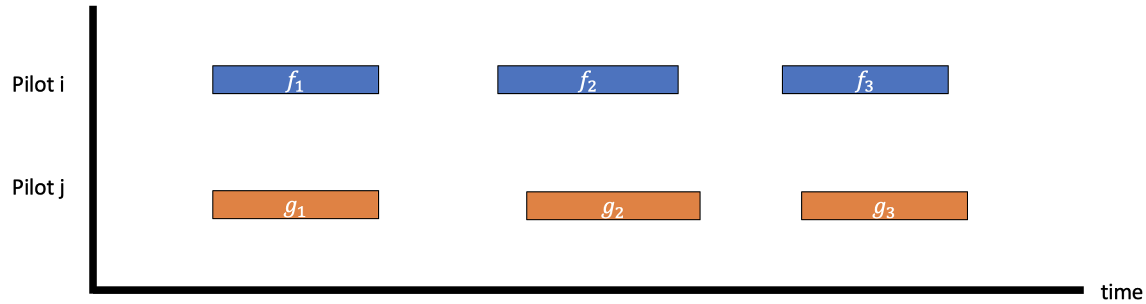

Figure 4-4: Example of schedule for pilot i and pilot j where pilot j could move-up to flight f_2.

overlap with f_2 and pilot j is not on leave at any point during f_2. Finally, Condition 2b) is satisfied because g_2 ends after f_2. Together, 2a) and 2b) guarantee that pilot j is available for the duration of f_2.

We define the following notation to formally describe the scenario.

S_f	set of valid flights that can source a move-up crew for flight f
O_f	set of flights that start before f but overlap with f
T_{move}	maximum allowable move-up time for flights to be included in S_f

S_f contains flights that start after f, but within $T_{\text{move}} = 5$ hours of d_f (i.e., $d_g > d_f + T_{\text{move}} \quad \forall g \in S_f$). We then define a binary variable $m_{if,jg}$ which is equal to 1 if pilot j assigned to flight g can move up to replace pilot i assigned to flight f. The following conditions must hold $\forall i \in I, f \in F, j \in I \mid Q_j \geq Q_i, g \in S_f$.

$$m_{if,jg} \leq x_{if} \tag{4.9}$$

$$m_{if,jg} \leq x_{jg} \tag{4.10}$$

$$m_{if,jg} \leq 1 - \sum_{o \in O_f} x_{jo} \tag{4.11}$$

Constraints 4.9 and 4.10 ensure that pilots i and j are assigned to flights f and g, respectively. Constraint 4.11 ensures pilot j is available for the start of f_2 and is not assigned to any flights that start before f_2 and overlap with f_2 (Condition 2a).

By construction, S_f guarantees that Condition 2b) is satisfied, as it contains flights with end times greater than or equal to f.

4.3.1 Swapping Crews

Up to this point, when $m_{if,jg} = 1$ we can only guarantee that pilot j can move up to pilot i's assignment on flight f. Ideally, pilot i and j would swap assignments such that pilot i is assigned to g and pilot j is assigned to f. In order for this swap to be possible, we need to more carefully consider downstream flights. We return to Figure 4-4. We want the following conditions to hold.

1. Pilot i is still available for their next scheduled assignment, f_3, if they are assigned to g_2

2. Pilot j is still available for their next scheduled assignment, g_3, if they are assigned to f_2

We define C_f as the set of flights that start after f and conflict with it. The following set of big-M constraints take care of the aforementioned conditions and must hold $\forall i \in I, f \in F, j \in I \mid Q_j \geq Q_i, g \in S_f$. These constraints would replace constraint 4.11.

$$\sum_{b \in C_f \setminus g} x_{jb} \geq 1 - M m_{if,jg} \tag{4.12}$$

$$1 - \epsilon \geq \sum_{b \in C_f \setminus g} x_{jb} - M(1 - m_{if,jg}) \tag{4.13}$$

$$\sum_{a \in C_g \setminus f} x_{fa} \geq 1 - M m_{if,jg} \tag{4.14}$$

$$1 - \epsilon \geq \sum_{a \in C_g \setminus f} x_{fa} - M(1 - m_{if,jg}) \tag{4.15}$$

Constraints 4.12 and 4.13 specify that if pilot j is assigned to a flight b (not including flight g) that conflicts with f, they cannot move-up to flight f, as they would not be able to make their next assignment. Similarly, constraints 4.14 and 4.15

ensure the equivalent constraint for pilot i. Note that these constraints alone do not guarantee that pilot i and j can swap assignments. First, we do not guarantee that pilot i is qualified for pilot j's assignment on g_2. However, this could be rectified with a decision variable that accounts for the required qualification satisfied (X_{ifq}). Second, if f_1 is delayed too much, then pilot i may not be able to fly on g_2 in Figure 4-4. But as long as $a_{f_1} + \text{MINSIT}_{f_1} \leq d_{g_2}$, pilot i should be available to swap with pilot j and fly on g_2.

4.4 Results

We now want to show the benefits of incorporating buffer time, expected delay propagation, and move-up crews. For move-up crews, we show results for the formulation in Section 4.3, which does not include swapping crew constraints. We use the same 10-week dataset as Chapter 2 and randomly generate disruptions, in the form of flight delays. Specifically, in each week, a proportion of flights get delayed by an hour amount defined by a uniform distribution between 1 and 18 hours. While some of these delay values would never appear in commercial aviation, they are possible in the Air Force where flight duration is very uncertain. As before, we schedule one week's worth of departures at a time.

We consider two options for handling these flight delays. In Option 1, when flight f' is delayed, schedulers delay all affected flights f until pilots from f' are ready to depart. Thus, delay propagates from f' to f. We do not consider cases where schedulers delay flights f more than strictly necessary because of f', nor do we consider cancellations. Option 1 could arise in practice when calling up pilots is avoided, like during a weekend or in the middle of the night. In Option 2, schedulers do not allow delay to propagate and instead utilize the disruption formulation in Chapter 3 to modify the original schedule. When flight f' is delayed, rather than delay downstream flights f as in Option 1, schedulers change assignments on f that are infeasible because of the delay to f'.

4.4.1 Option 1: Propagate Delays with No Reassignments

We compare the performance of the overqualification and training requirements objective ("Baseline") with the Baseline + buffer times and delay propagation objective ("BDP"). The move-up formulation is not relevant, since with Option 1 we do not consider reassignments. To see the effects of the number of delayed flights, we vary the delay fraction ("DF"), which is the fraction of flights that have a random delay. Table 4.4.1 displays summary statistics for the two objectives with four delay fraction values. The "total buffer" is calculated in the original weekly schedule before any flight are delayed. It sums the time between successive flight assignments of the same pilot. If a pilot is only assigned one flight in the time period, the buffer time is the end of the time period minus the arrival time of the flight that they are assigned to. We also report the number of pairs of flights with a buffer time less than five hours, as these are tight connections that BDP tries to avoid. The initial delay randomly generated is shown in the "NPD", or non-propagated delay, column. The resulting propagated delay is shown in the "PD" column.

Table 4.1: Comparison of Baseline vs. BDP (model incorporating buffer and delay propagation) with Option 1. NPD is initial non-propagated delay, PD is propagated delay.

	Delay Frac.	Total Buffer (h)	Buffer < 5 hours (flts.)	Total Delay (h)	NPD (h)	PD (h)
Baseline	0.25	23,012	191	2,361	1,680	681
BDP	0.25	27,694	1	1,778	1,680	98
Baseline	0.50	22,633	190	5,907	3,552	2,355
BDP	0.50	26,256	0	3,835	3,552	283
Baseline	0.75	23,622	187	9,529	5,448	4,081
BDP	0.75	26,969	0	6,240	5,448	792
Baseline	1.0	24,380	185	13,255	7,224	6,031
BDP	1.0	27,024	1	8,221	7,224	997

We first consider the total buffer and number of flights with a tight buffer. Compared to the baseline, the BDP consistently has higher total buffer than the baseline. Moreover, the baseline has 185-191 flight pairs with a buffer less than five hours, compared to one or fewer with the BDP. With more buffer in the schedule, we expect the BDP to perform better than the baseline when faced with flight delays. Note that for a given delay fraction value, the baseline and BDP models encountered the same set of pseudorandom flight delays (hence, their NPDs are the same). For all delay fractions, the BDP experiences less propagated delay than the baseline. With delay fraction of 1, the baseline has over 5,000 hours more propagated delay than the baseline. The gap between baseline propagated delay and BDP propagated delay widens as delay fraction increases.

We further explore the nature of the propagated delay experienced by considering what we call the *degrees of propagation*. When the randomly generated delay of Flight A propagates to Flight B, we consider this first-degree delay propagation. If Flight B in turn delays Flight C as a result of the delay to Flight A, this is second-degree delay propagation, and so on. Note that since the model is only aware of one week's worth of departures at a time, delay propagation does not span between flights departing in different weeks. Figure 4-5 shows a histogram of degrees of propagation for the baseline and BDP across the four delay fractions. As degrees of propagation increases, the frequency of occurrence decreases. As expected, the maximum degree of propagation seen with the BDP is lower than with the baseline. At delay fraction of 0.5, the baseline experiences at most 10 degrees of delay propagation, whereas the BDP sees at most 4. Overall, the BDP also sees fewer flights with propagated delay than the baseline in each degree of propagation.

The difference between the baseline and BDP is even more pronounced when looking at the magnitude of delay propagation. Figure 4-6 is similar to Figure 4-5 but displays the total delay propagation for each degree of propagation. With the BDP, the propagated delay for a given degree of propagation never exceeds 400 hours. In contrast, with delay fraction of 1, the baseline propagated delay is over 1400 hours for first-degree delay propagation. Such high delay propagation values would significantly disrupt operations, as carefully planned flight schedules would significantly change.

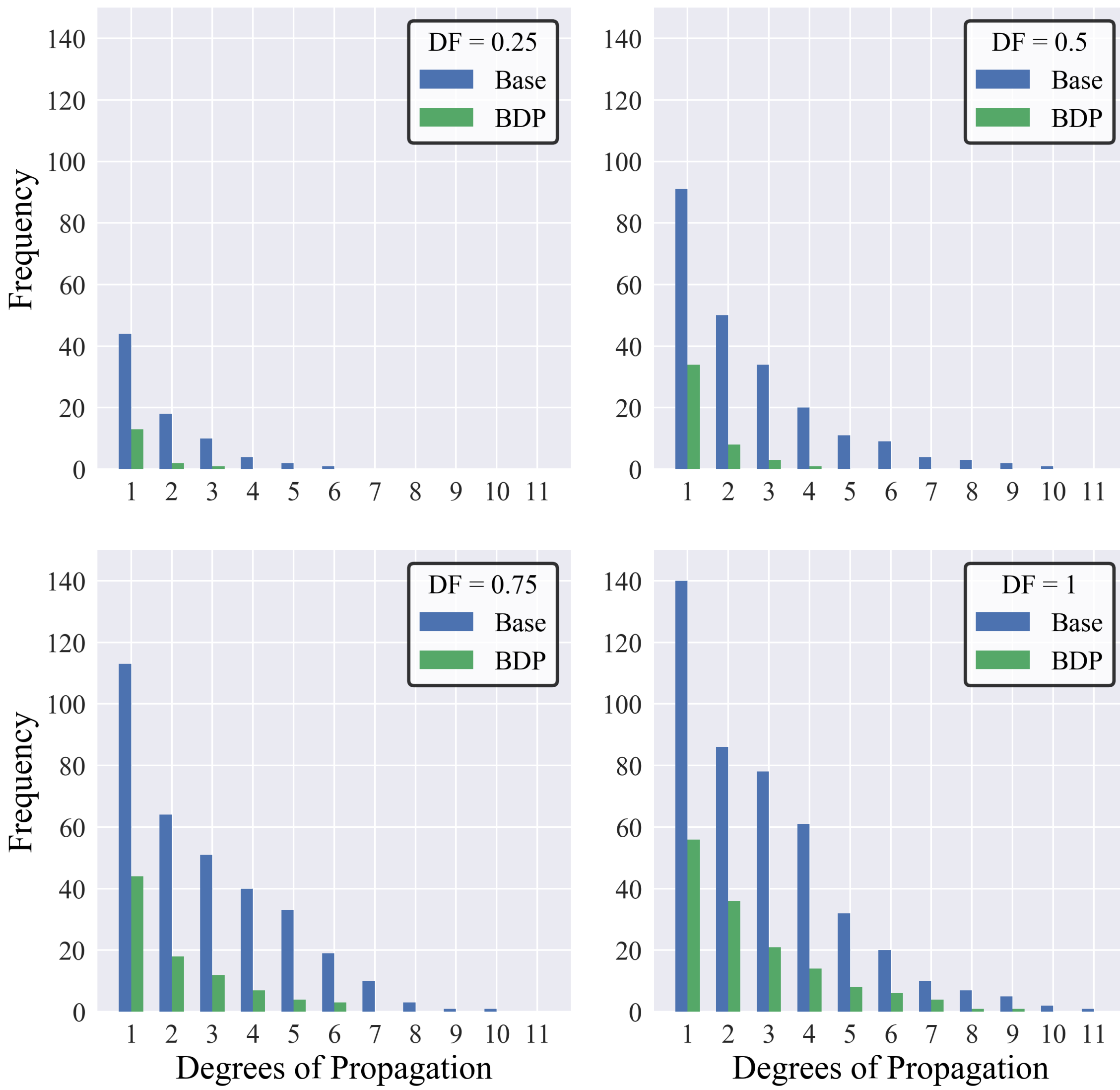

Figure 4-5: Frequency of degrees of delay propagation. DF (delay fraction) indicates the fraction of flights delayed.

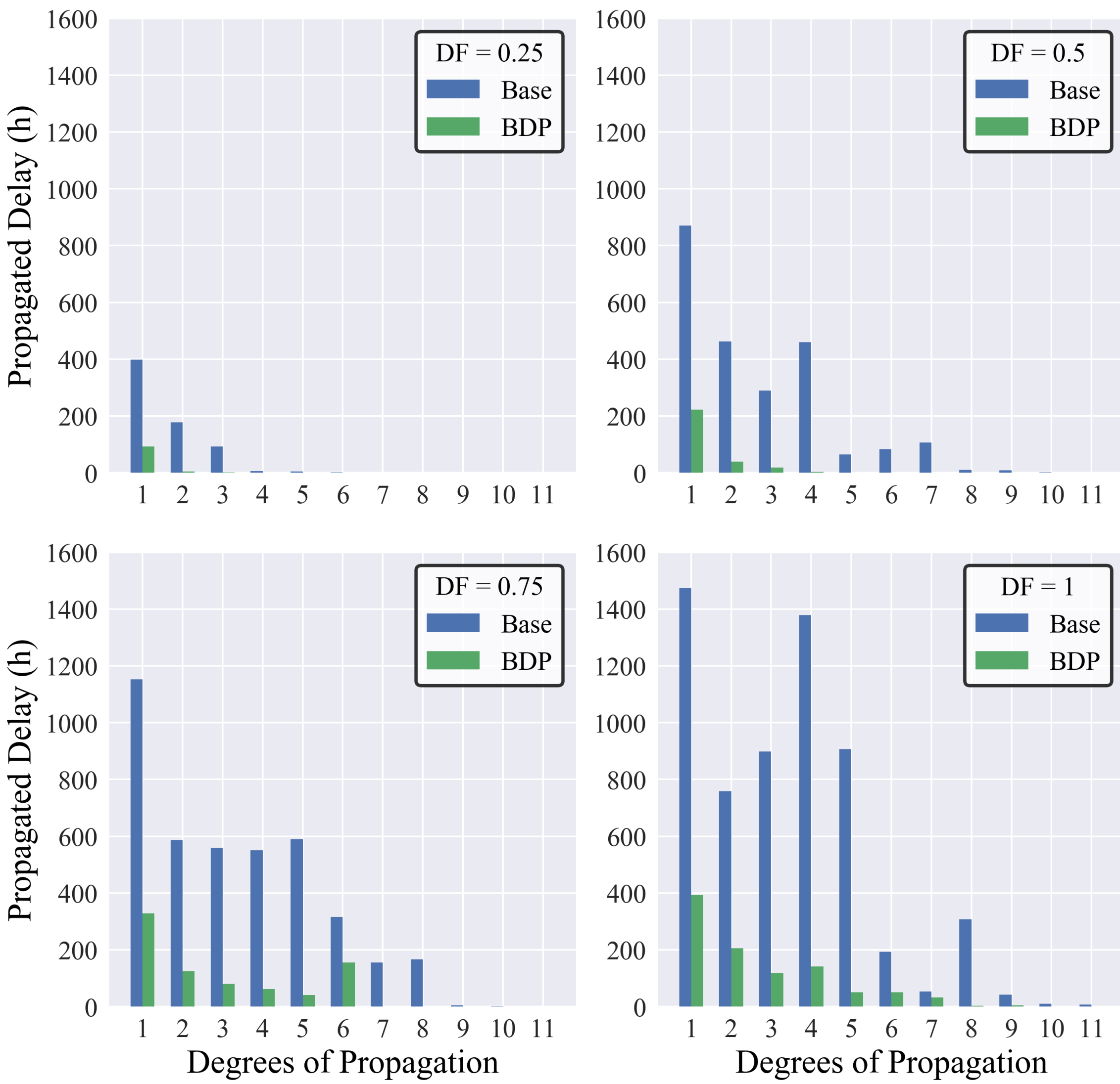

Figure 4-6: Total delay for different degrees of delay propagation. DF (delay fraction) indicates the fraction of flights delayed.

4.4.2 Option 2: Handle Disruption with Reassignments

With Option 2, schedulers utilize the disruption formulation presented in Chapter 3 to determine reassignments to handle disruptions. We continue to use late-arriving flights as the disruption example. We evaluate three models on Option 2: baseline, BDP, and BDP with move-up crews incorporated ("BDP+Move-up"). We expect that the BDP formulation will lead to fewer reassignments being necessary, as we saw with Option 1 that delay is less likely to propagate with the BDP formulation. Further, we expect that the BDP+Move-up model will increase the frequency of move-up reassignments and reduce the number of call-ups. We first define several counting and time metrics that will be used to compare the models.

Counting Metrics

We first define three counting metrics. Figure 4-7 shows four subplots with a) being the original schedule and b), c), and d) showing examples of call-ups, move-ups, and swaps, respectively. In the original schedule, pilot i is assigned to f_1; pilot j is assigned to f_2; and pilot k is unassigned. Suppose that pilot i can no longer be assigned to f_1 because of a late arriving preceding flight. We consider three reassignment options below.

1. **Call-ups** (Figure 4-7 b): With a call-up, we recruit a unassigned pilot to fill-in for the disruption. Pilot k, who was previously unassigned, is assigned to f_1. Pilot i is now unassigned for the given time period. Pilot j is unaffected.

2. **Move-ups** (Figure 4-7 c): With a move-up, we choose a pilot currently assigned to a flight later than f_1 to replace pilot i. Here, pilot j moves up to f_1. Note that in this example, pilot k is then called up to substitute for pilot j. Move-ups typically necessitate call-ups to fill in new vacancies; however, there could be be a chain of several move-ups, followed by one call-up.

3. **Swaps** (Figure 4-7 d): Suppose that pilot i is unavailable for f_1, but is qualified and available for f_2. Then, we could perform a swap in assignments between

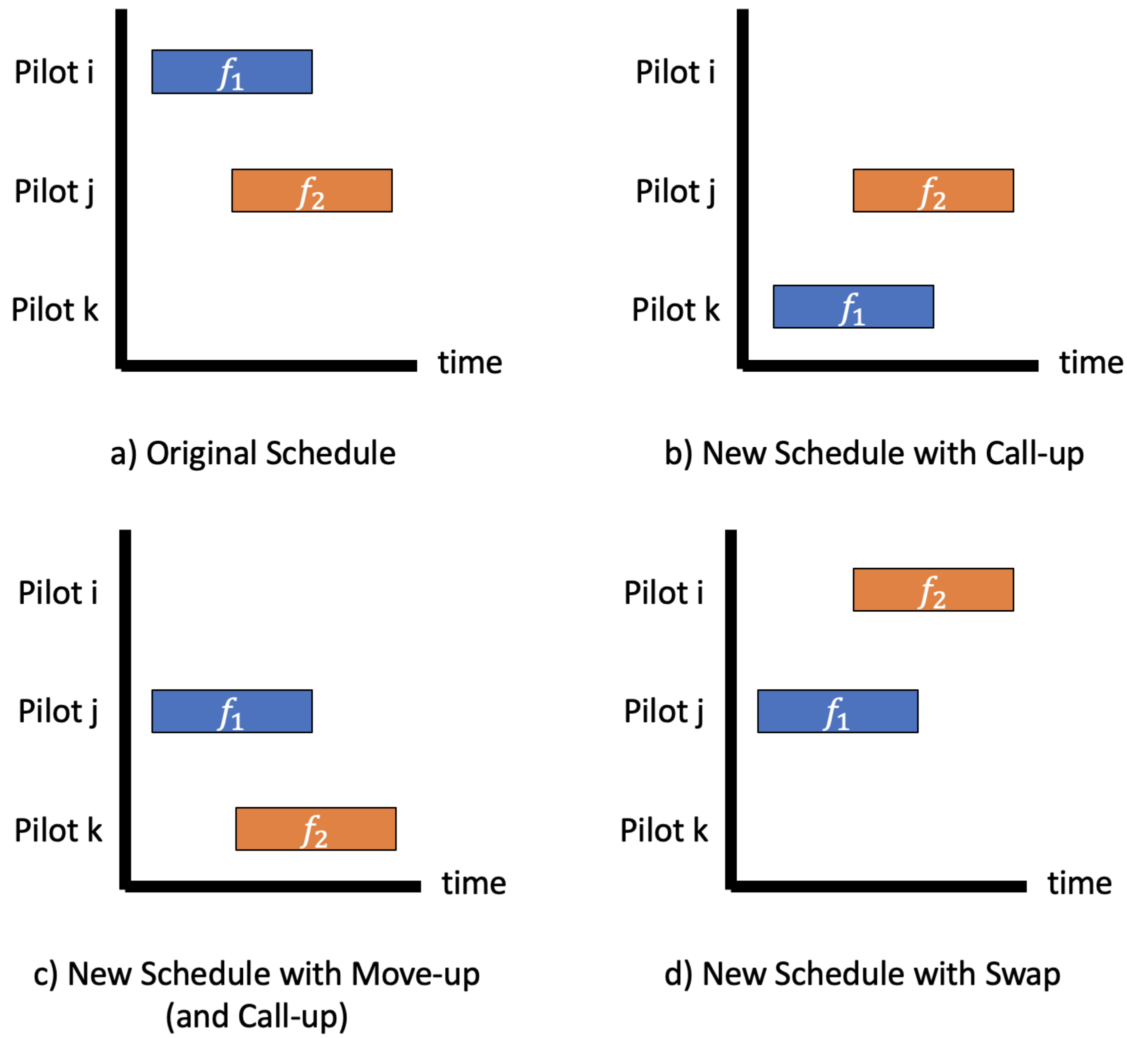

Figure 4-7: Examples of call-ups, move-ups, and swaps with respect to original schedule in a), given that pilot i can no longer be on f_1.

pilot i and j. This has the benefit of being the only reassignment option that does not involve pilot k.

Time Metrics:

While the counting metrics are intuitive, they do not consider the extra time burden placed on pilots. In particular, two different move-ups could have very different time burdens. For most pilots, moving up to a one-hour simulator in lieu of a three day mission that starts at the same time would not be an issue. On the other hand, the reverse (moving up to a three day mission from a one-hour simulator) would be very disruptive.

To this end, we introduce two time metrics that classify the hours of reassignment into two categories: "switch assignment" hours and "new assignment" hours. Figure 4-8 shows an original schedule on the left and a new schedule with a swap on the right.

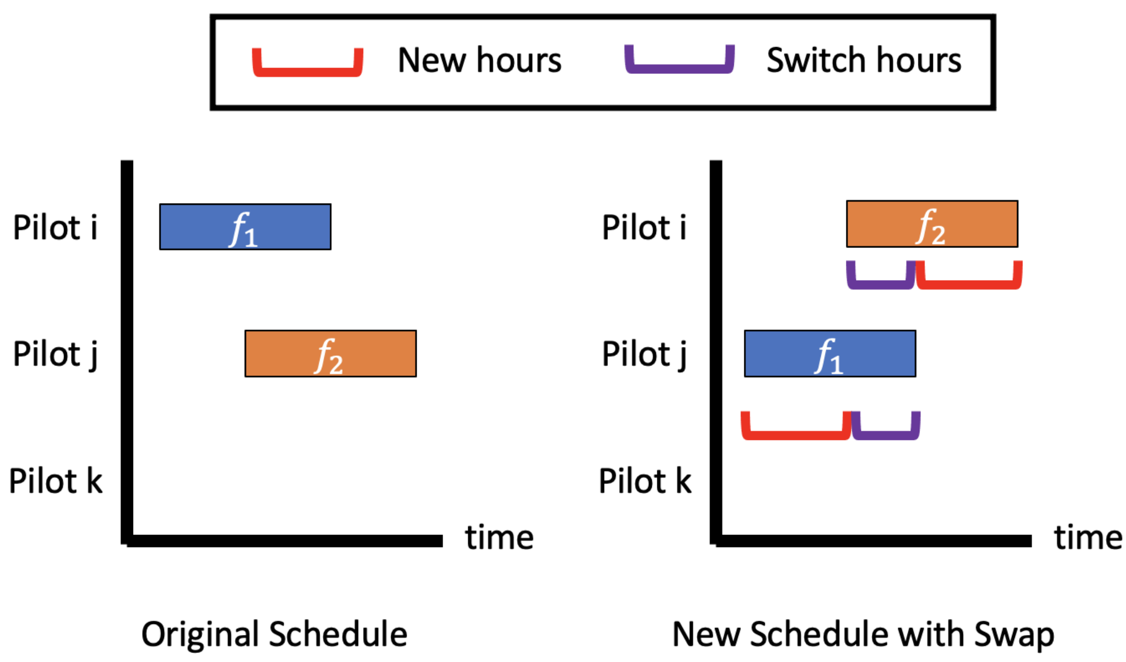

Figure 4-8: Examples of switch assignment hours and new assignment hours for a swap between pilot i and j.

1. **Switch assignment hours**: These are the hours in which pilots were assigned to a flight in the original schedule, but are now assigned to a different flight. In Section 4.3, we did not penalize switch assignment hours because we expect that since they were going to be active anyway, pilots do not experience too much disutility from this. In Figure 4-8, pilot i and j have switch assignment hours between d_{f_2} (scheduled departure time of f_2) and a_{f_1} (scheduled arrival time of f_1).

2. **New active hours**: These are the hours in which pilots are newly active, since they were not assigned to any flights during this time period in the original schedule. Pilot j has new active hours between d_{f_1} and d_{f_2}, while pilot i has new active hours between a_{f_1} and a_{f_2}. These are burdensome for pilots as this may disrupt their personal plans. However, note that since we account for pilot leave and TDY, pilots should be available during these new active hours.

Table 4.4.2 presents the counting and time metrics across two delay fraction values for the three objectives. We see that the baseline requires the most reassignment, given that it has the highest number of call-ups, move-ups, new active hours, and switch assignment hours. This is expected given that the baseline schedule does not have as much slack as the BDP or BDP+Move-up, so it is forced to make more reassignments to avoid delay propagation.

Table 4.2: Comparison of counting and time metrics with delay fraction of 0.25 or 0.50 with Option 2.

	Delay Frac.	Call-ups	Move-ups	Swaps	New Active Hours	Switch Hours
Baseline	0.25	148	25	1	3,280	2,323
BDP	0.25	8	2	0	412	280
BDP+Move-up	0.25	4	5	5	334	506
Baseline	0.50	238	36	1	4,968	2,823
BDP	0.50	12	4	1	654	379
BDP+Move-up	0.50	9	14	5	538	880

While baseline is clearly worse than the other two, there is a trade-off between BDP and BDP+Move-up. BDP has fewer number of reassignments and time duration of schedule changes than BDP+Move-up. The BDP+Move-up requires more reassignments than BDP, but more of these reassignments are move-ups and swaps than with the BDP. A scheduler who is most concerned with minimizing the summation of the counting metrics or the time metrics may choose BDP. However, a scheduler who is more concerned with minimizing call-ups and new active hours may prefer BDP+Move-up. Both of these scenarios depend on the relative preference between call-ups and move-ups/swaps. An additional consideration is computational complexity. While the baseline and BDP solve within a matter of seconds, the BDP+Move-up took an average of 15 minutes to solve a week's schedule. This is due to the large number of variables needed to track the possible permutations of move-up crews.

Chapter 5

Conclusion

In this thesis, we analyzed disruptions and robustness in Air Force crew scheduling. We first discussed the key similarities and differences between Air Force crew scheduling and airline crew scheduling. We then presented the baseline integer program and discussed trade-offs between two objectives: overqualification and training requirements. While the baseline integer program is useful for "clean-slate" scheduling, we developed a disruption formulation to modify previously generated schedules in the face of disruptions. We developed schedule visualization tools for a set of user interviews and discussed some main takeaways. Finally, we adapted two notions of robustness from commercial aviation: 1) buffer time and expected delay propagation and 2) move-up crews. We showed how these formulations handle disruptions better than the baseline formulation. We showed results with two alternative strategies for handling disruptions: without reassignments (allows delay to propagate but less disruptive to schedule) and with reassignments (avoids delay propagation but potentially more disruptive). Without reassignment, incorporating buffer time and expected delay propagation blunts the frequency and magnitude of delay propagation. With reassignment, incorporating buffer time and move-up crews reduce propagated delay with trade-offs in number of reassignments and type of reassignments. Certain reassignment types like call-ups are more disruptive than others like move-ups or swaps.

There are several areas of potential future work. When handling disruptions, schedulers do not pick one strategy to use for the entire scheduling window. We could develop a model that blends the two strategies in Sections 4.4.1 and 4.4.2 and considers the trade-off between the cost of delay propagation and the cost of calling-up/reassigning crew. In the face of disruptions, it may also be desirable to adjust the scheduled times of downstream flights. Thus, we could further enhance the disruption-handling model by considering limited schedule adjustments. This is a more tractable problem than the full-fledged joint flight and crew scheduling problem discussed in Section 1.3. Finally, other members of our research team are exploring reinforcement learning-based approaches for scheduling. These approaches may be more scalable and adept at predicting patterns for more intelligent scheduling.

The *Puckboard* application has been deployed across over 125 squadrons. Users are already benefiting from the centralized housing of data on pilot qualifications, training requirements, and availability. The next step is to integrate our scheduling formulations into *Puckboard* interface. Once this is done, schedulers will be able to generate schedules more efficiently. Moreover, the schedules they generate will ideally be more desirable in terms of overqualification, training requirements, and robustness. This has the potential to streamline Air Force operations and improve pilot quality of life.

Bibliography

[1] Claudine Biova Agbokou. Robust Airline Schedule Planning: Review and Development of Optimization Approaches.

[2] Cynthia Barnhart, Amy M. Cohn, Ellis L. Johnson, Diego Klabjan, George L. Nemhauser, and Pamela H. Vance. *Airline Crew Scheduling*.

[3] Valentina Cacchiani, Alberto Caprara, Laura Galli, Leo Kroon, Gabor Maroti, and Paolo Toth. Recoverable Robustness for Railway Rolling Stock Planning.

[4] Avishai Ceder. Urban Transit Scheduling: Framework, Review and Examples. *Journal of Urban Planning and Development*, 128(4):225–244, December 2002.

[5] Jacob Feldman, Nan Liu, Huseyin Topaloglu, and Serhan Ziya. Appointment Scheduling Under Patient Preference and No-Show Behavior. *Operations Research*, 62(4):794–811, August 2014.

[6] Roger S Jacobs. Optimization of daily flight training schedules. Master's thesis, Naval Postgraduate School, 2014.

[7] Matthew Koch. Air Force Crew Scheduling: An Integer Optimization Approach. Master's thesis, Massachusetts Institute of Technology, 2021.

[8] Peter J. M. van Laarhoven, Emile H. L. Aarts, and Jan Karel Lenstra. Job Shop Scheduling by Simulated Annealing. *Operations Research*, 40(1):113–125, 1992. Publisher: INFORMS.

[9] D Macey. Fallon Range Training Complex Flight Event Timetabling. Master's thesis, Naval Postgraduate School, 2017.

[10] Anne Mercier and François Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34(8):2251–2265, August 2007.

[11] Mai-Ha Phan and Kap Hwan Kim. Collaborative truck scheduling and appointments for trucking companies and container terminals. *Transportation Research Part B: Methodological*, 86:37–50, April 2016.

[12] Sergey Shebalov and Diego Klabjan. Robust Airline Crew Pairing: Move-up Crews. *Transportation Science*, 40(3):300–312, August 2006.

[13] Robert J Slye. Optimizing Training Event Schedules at Naval Air Station Fallon. Master's thesis, Naval Postgraduate School, 2018.

[14] Mirela Stojković and François Soumis. An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem. *Management Science*, 47(9):1290–1305, September 2001.

[15] Oliver Weide, David Ryan, and Matthias Ehrgott. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 37(5):833–844, May 2010.

[16] Xin Wen, Hoi-Lam Ma, Sai-Ho Chung, and Waqar Ahmed Khan. Robust airline crew scheduling with flight flying time variability. *Transportation Research Part E: Logistics and Transportation Review*, 144:102132, December 2020.

[17] T.C. Wong, M. Xu, and K.S. Chin. A two-stage heuristic approach for nurse scheduling problem: A case study in an emergency department. *Computers & Operations Research*, 51:99–110, November 2014.