NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPROVING AUTOMATED SCHEDULES FOR NAVAL AIR STATION KINGSVILLE

by

Jasmine L. Ye

March 2021

Thesis Advisor:	Robert F. Dell
Co-Advisor:	Emily M. Craparo
Second Reader:	Jeffrey F. Hyink

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE IMPROVING AUTOMATED SCHEDULES FOR NAVAL AIR STATION KINGSVILLE			**5. FUNDING NUMBERS**
6. AUTHOR(S) Jasmine L. Ye			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Currently, many squadrons in the Naval Aviation community handwrite their daily flight schedules, which is typically an all-day effort. This thesis creates an optimization model to build schedules computationally instead of manually for Navy’s Training Squadron 22 (VT-22), which specializes in Intermediate Jet and Advanced Strike training. An optimized scheduling process can improve the efficiency of the training pipeline, saving money and improving aviation readiness.

A preliminary model, Training Event Scheduling Tool (TEST), was provided to VT-22 in 2019 by Meditz. TEST takes a spreadsheet containing student prerequisites, instructors, and events, and creates a daily or weekly schedule at an hourly resolution. This thesis formulates and tests a revised integer linear program, TEST-2, an enhancement to TEST that models weather, substitutable events, and student currency. TEST-2 creates daily schedules in less than 10 minutes and weekly schedules in about four hours. These schedules consider a majority of the necessary constraints for a useable schedule. For a sample week’s input provided by VT-22, TEST-2 schedules about 60 more events over the course of the week than were manually scheduled and completed. Currently, many events are cancelled due to instructor non-availabilities, weather, and jet availability. Because TEST-2 considers these three factors in building its schedules, cancellations due to these factors are minimized.

14. SUBJECT TERMS			**15. NUMBER OF PAGES**
optimization, naval aviation, scheduling, Training Event Scheduling Tool, TEST, VT-22			63
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**IMPROVING AUTOMATED SCHEDULES FOR
NAVAL AIR STATION KINGSVILLE**

Jasmine L. Ye
Ensign, United States Navy
BS, United States Naval Academy, 2019

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
March 2021**

Approved by: Robert F. Dell
Advisor

Emily M. Craparo
Co-Advisor

Jeffrey F. Hyink
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Currently, many squadrons in the Naval Aviation community handwrite their daily flight schedules, which is typically an all-day effort. This thesis creates an optimization model to build schedules computationally instead of manually for Navy's Training Squadron 22 (VT-22), which specializes in Intermediate Jet and Advanced Strike training. An optimized scheduling process can improve the efficiency of the training pipeline, saving money and improving aviation readiness.

A preliminary model, Training Event Scheduling Tool (TEST), was provided to VT-22 in 2019 by Meditz. TEST takes a spreadsheet containing student prerequisites, instructors, and events, and creates a daily or weekly schedule at an hourly resolution. This thesis formulates and tests a revised integer linear program, TEST-2, an enhancement to TEST that models weather, substitutable events, and student currency. TEST-2 creates daily schedules in less than 10 minutes and weekly schedules in about four hours. These schedules consider a majority of the necessary constraints for a useable schedule. For a sample week's input provided by VT-22, TEST-2 schedules about 60 more events over the course of the week than were manually scheduled and completed. Currently, many events are cancelled due to instructor non-availabilities, weather, and jet availability. Because TEST-2 considers these three factors in building its schedules, cancellations due to these factors are minimized.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AGL	above ground level
CNATRA	Chief of Naval Air Training
CQ	carrier qualification
FIST	Flight Instructor Standardization and Training
FRS	Fleet Replacement Squadron
FTS	Flight Training Scheduler
NAS	naval air station
NIFE	Naval Introductory Flight Evaluation
SAT	Scheduling Assistance Tool
SKEDSO	scheduling officer
SNA	student naval aviator
T-45	McDonnell Douglas T-45 Goshawk (operated by U.S. Navy)
TEST	Training Event Scheduling Tool
TEST-2	Training Event Scheduling Tool Version 2.0
TSHARP	Training Command Sierra Hotel Aviation Readiness Program
VT-22	Training Squadron Twenty-Two

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Naval Aviation training pipeline has a backlog. Many students are idle for weeks or months between training phases and overall time-to-train in certain phases is longer than allocated. This backlog exists in many of the phases of training; for this thesis effort, we focus on reducing the backlog in the Intermediate and Advanced phases of Strike training, also known as the jet pipeline. Training Squadron 22 (VT-22) is one of the Navy's strike training squadrons, and they currently produce their daily flight schedules by hand. This is typically an all-day effort and requires the attention of the squadron military Scheduling Officer (SKEDSO) and two civilian schedulers.

VT-22 requested a tool that builds schedules computationally instead of by hand. Along with saving time for the schedulers, an optimization-based schedule has the potential to contain more events and to be more robust to cancellations due to factors such as weather. Meditz (2019), in a prior thesis effort, created the initial model called the Training Event Scheduling Tool (TEST). The TEST prototype considers a great number of the constraints necessary for a flight schedule, but not everything. This led VT-22 to request an improved version of TEST. This thesis extends TEST by creating TEST-2 with additional constraints to build more useable schedules.

The overall goal of this thesis work is to create a useable tool for VT-22 that schedules more events than would be scheduled by hand and reduces overall time-to-train. Some secondary goals stem from this main goal. The first is to minimize warmup events, which are repeated events due to a student falling out of a currency window. The second is to maintain a given number of flights a week for each student. The third goal is to create a robust schedule that considers external factors such as weather. Along with considering these goals, this tool must also consider numerous scheduling constraints.

TEST includes numerous scheduling constraints already, including instructor and student per-day event limits, maximum lecture hours allowed in one day, aircraft availability, crew rest rules, prerequisite events, and special event requirements. The three main constraints VT-22 wished to be added to the model to make it more useable

are event weather requirements, student currency consideration, and combined flight events.

With regard to flight event weather requirements, each flight event in the syllabus has a minimum ceiling and visibility requirement in order to be flown. TEST-2 incorporates weather forecast information in two ways. First, the model does not allow events to be flown in time periods that do not meet the minimum ceiling and visibility requirements. Second, we build a reward structure to prioritize periods with forecasted good weather for events with stricter weather requirements.

To incorporate student currency into the model, we need to know when the student last completed prerequisite events. If the student completed a prerequisite within a certain window of days, the student is considered "current" and can be scheduled for the follow-on event. If they are outside of the "currency window," they must do a warmup event, which often involves repeating the prerequisite event or a similar event before they can be scheduled for the next event in the syllabus. VT-22 wishes to reduce the number of warmup events because students who fall out of currency are stagnant in the training pipeline. Also, students doing warmup events take away resources, jets, and instructors that could be used to progress other students in the syllabus. TEST-2 uses a reward structure to prioritize students who are close to falling out of currency. Additionally, a smaller reward is used to bring students back into currency who have already fallen out of currency and are in a "warmup window."

Schedulers have the ability to utilize what are known as out-and-in events and in-and-in events while scheduling. These events are two events combined together and scheduled as one. The difference between out-and-in and in-and-in events is that out-and-in events can land at a non-local airport in between, but in-and-in events must stay at the home airport. VT-22 prefers to use these combined events because briefs can be combined and this reduces overall time scheduled. To implement this in TEST-2, we created a set of substitutable events. TEST-2 rewards scheduling combined flight events with the goal of saving scheduled time and reducing overall time-to-train.

Considering these three new constraints, we built TEST-2 in both a daily and weekly version. The daily model runs in about 10 minutes while the weekly model runs in about 4 hours. TEST-2 is an integer linear program that is run in Python using Pyomo and an integer linear programming solver. To test our model, VT-22 collected data from November 2–6, 2020. This data includes student and instructor non-availabilities, the posted schedule, the actual schedule results due to cancelled events, the weather for the week, instructor qualifications, and student currency data. From this, we built a new dataset and ran it through TEST-2. TEST-2 schedules about 60 more events than were actually completed by VT-22 for the week of November 2–6. In all three categories, lectures, simulators, and flight events, TEST-2 schedules more events and, in particular, 30 of the 60 events are flight events.

TEST-2 provides useable schedules to VT-22 that are optimized and robust. In about ten minutes, the daily model provides a schedule that considers weather forecast, student currency, combined flight events, and many other constraints, rather than requiring a full day of effort from multiple schedulers. In about four hours, the weekly model creates a schedule that considers a majority of the constraints for a useable schedule that helps VT-22's schedulers have an outlook for the week ahead. Implementing TEST-2 will help VT-22 create schedules that incorporate more events in all categories, which can reduce time-to-train.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Robert Dell and Dr. Emily Craparo, for all of their help on this thesis and throughout my time at NPS. Thank you both for dedicating time out of your busy lives to help me with writing, coding, and just checking in with me to see how I am doing. I would also like to thank my second reader, CAPT Jeffrey Hyink, for his support in this thesis effort.

Additionally, I want to thank the team at VT-22 who have supported this thesis, particularly LCDR Scott Simpson. Thank you for being available always to answer all of my questions, however big or small.

Finally, thank you to my friends and family who have supported me throughout my time in Monterey. Thank you for providing much-needed stress relief, support, and fun this past year-and-a-half.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Just over one hundred years ago, Eugene Ely was the first pilot to successfully land and take off from a U.S. Navy ship. With this event, naval aviation was born. Since then, naval aviation has grown and developed to encompass a corps of highly trained professionals. Each student naval aviator (SNA) goes through a rigorous training program, about two years in length, before receiving their "Wings of Gold." SNAs begin their flight training in Pensacola, Florida. At Naval Air Station (NAS) Pensacola, SNAs must pass Naval Introductory Flight Evaluation (NIFE). After passing all of their flights and exams in NIFE, SNAs move on to Primary training. Over the course of approximately six months in Primary, instructors evaluate each student on their performance in various stages of flying. At the end of Primary, based on instructor scores and individual SNA preferences, the Navy assigns a flying platform to each student. Jets are one of the most coveted platforms in Naval Aviation. For those SNAs selected to fly jets, the next phases of training are Intermediate and Advanced, which take place in either NAS Kingsville or NAS Meridian. The syllabus for Intermediate and Advanced is comprised of numerous classroom, simulator, and flight events. For jet training, also known as the Strike pipeline, these events take about one year to complete. Upon completion of Advanced, the Navy awards student aviators with their "Wings of Gold." For the final portion of training, the newly designated aviators report to their assigned fleet replacement squadron (FRS). Following training at the FRS, aviators report to their fleet squadron and their training is complete.

In Primary, Intermediate, and Advanced, SNAs are assigned to a training air wing. Each training air wing is made up of multiple training squadrons. In a squadron, one officer is typically in charge of scheduling events for all of the SNAs and instructors. This officer is known as the scheduling officer (SKEDSO). Currently, in most squadrons, the SKEDSO creates daily schedules by hand. This can be very complicated, as there are many factors to take into account while building a schedule, such as weather, student and instructor qualifications, and number of planes available. One of the training squadrons at

NAS Kingsville, Training Squadron Twenty-Two (VT-22), recognized that the current process for creating schedules can be improved upon using optimization. Meditz (2019) created a prototype tool, the Training Event Scheduling Tool (TEST) for VT-22. TEST is an integer linear program that creates hourly schedules for classroom, simulator, and flight events for the students and instructors in VT-22. Although TEST satisfies VT-22's basic goals for a scheduling tool, officers at VT-22 wish to incorporate other aspects of scheduling into an improved tool. This thesis develops TEST-2, an improvement upon TEST, which takes into consideration factors such as weather, student currency, and combined flight events. TEST-2 outputs both a daily schedule and weekly schedule that adhere to the training requirements of the Strike syllabus.

B. SYLLABUS

Students who have been selected to fly jets report either to Training Air Wing One at NAS Meridian or Training Air Wing Two at NAS Kingsville. Both training wings follow the same syllabus to educate and graduate SNAs to the FRS. Within each training air wing are two training squadrons. This thesis focuses specifically on Training Squadron Twenty-Two at NAS Kingsville. The work, however, is relevant to all four training squadrons.

VT-22 is composed of about 75–80 students and 35 instructors. Students arrive to VT-22 throughout the year and as a result, they are at different places in the syllabus on a given week. The goal of VT-22's schedulers is to graduate SNAs with minimal time-to-train. Additionally, students should be treated equally with respect to scheduling opportunities. One student should not be moved through the syllabus quicker than another if it can be avoided.

Students and instructors generally partake in events Monday through Friday all year, with the exception of federal holidays and a winter leave period. Saturday flights can occur, but schedulers try to avoid using this day if possible. Chief of Naval Air Training (CNATRA) states that student time-to-train for Intermediate at TW-2 should be 120.4 training days which equals 26.5 calendar weeks. Advanced time-to-train is supposed to be 111.6 training days or 24.8 calendar weeks (Naval Air Training

Command 2019). In total, the Strike training at VT-22 should take 232 training days or 51.3 calendar weeks, approximately one year. These estimates take into consideration holidays, weekends, safety standdowns, and any other expected nonworking days.

Throughout the syllabus, students must complete multiple lecture, simulator, and flight events. The flight events are conducted in the McDonnell Douglas T-45 Goshawk (T-45) trainer. Both the Intermediate and Advanced syllabi are composed of multiple stages. Within the Intermediate phase, there are 18 main flight support stages and in Advanced, there are 10 stages. These stages and their corresponding hours are displayed in Figure 1 and Figure 2.

INTERMEDIATE JET FLIGHT SUPPORT		
Stage	**Symbol**	**Hours**
Crew Resource Management	CRM11	3.0
Operational Risk Management	ORM11	1.0
NACES Flight Physiology	SEA11	3.0
Cockpit Orientation	CO11	7.3
Emergency Procedures	EP11	14.5
BI/RI Course Rules	CR11	1.0
Course Rules	CR12	3.0
Familiarization Flight Procedures	FAM11	8.5
Out-of-Control Flight (OCF) Procedures	OCF11	3.0
NATOPS/NATOPS Examinations	NA11	6.0
Night Familiarization Flight Procedures	NFM11	3.5
Basic Instrument Flight Procedures	BI11	10.5
Radio Instrument Flight Procedures	RI11	8.5
Airways Navigation Flight Procedures	AN11	2.0
Instrument Rating Flight Procedures	IR11	4.0
Section Formation Flight Procedures	FRM11	5.5
Division Formation Flight Procedures	DIV11	2.5
Field Carrier Landing Flight Procedures	FCL11	2.5
Total		**89.3**

Figure 1. Intermediate Strike Flight Stages.
Source: Naval Air Training Command (2019).

ADVANCED STRIKE FLIGHT SUPPORT		
Stage	**Symbol**	**Hours**
Operational Navigation Flight Procedures	ON11	3.7
Section Low-Level Flight Procedures	ON12	2.5
Road Recce Flight Procedures	RR11	2.5
Strike Flight Procedures	STK11	5.0
Night Formation Flight Procedures	NFR11	2.2
Tactical Formation Flight Procedures	TAC11	4.0
Basic Tactical Formation	TAC12	2.5
1 V 1 Basic Fighter Maneuvering Flight Procedures	BFM11	5.8
2 V 1 Section Engaged Maneuvering Flight Procedures	SEM11	3.7
Carrier Qualification Landing Flight Procedures	CQL11	6.0
Total		**37.9**

Figure 2. Advanced Strike Flight Stages.
Source: Naval Air Training Command (2019).

Certain stages are prerequisites and must be completed before advancing to other stages. Within each stage, schedulers must also adhere to prerequisites when assigning students to events. A visual representation of the Intermediate flight and Advanced Strike complete flow are displayed in Figure 3 and Figure 4, respectively.

C. SCHEDULING REQUIREMENTS

To build a useable schedule, the SKEDSO must consider many factors, including crew rest, number of aircraft available, weather, student currency, and combined flight events. While TEST considers many of these constraints in its formulation, not all of them are included. TEST-2 improves upon TEST by incorporating three additional central constraints that allow the schedules generated to be more useable. The factors considered in TEST and those three main constraints are described in this section.

1. Scheduling Factors Considered in TEST

Many different scheduling factors are considered in TEST and built into the model using various constraints. The factors included are crew day, instructor qualifications, student and instructor non-availabilities, night flights, prerequisites, jet availability, on-wing instructors, simulator capacities, and carrier qualification (CQ) specifications (Meditz 2019).

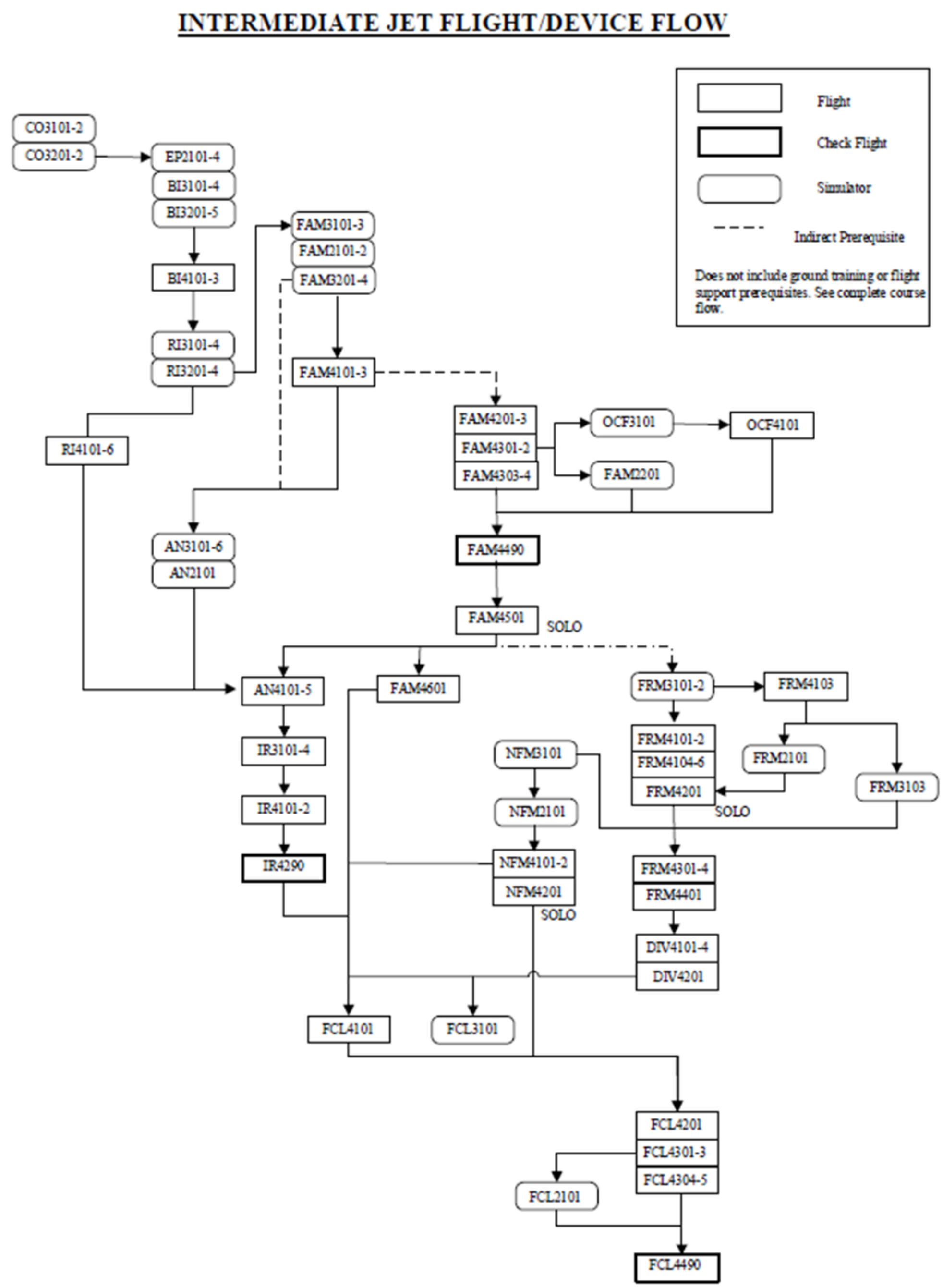

Figure 3. Intermediate Strike Flight Flow. Source: Naval Air Training Command (2019).

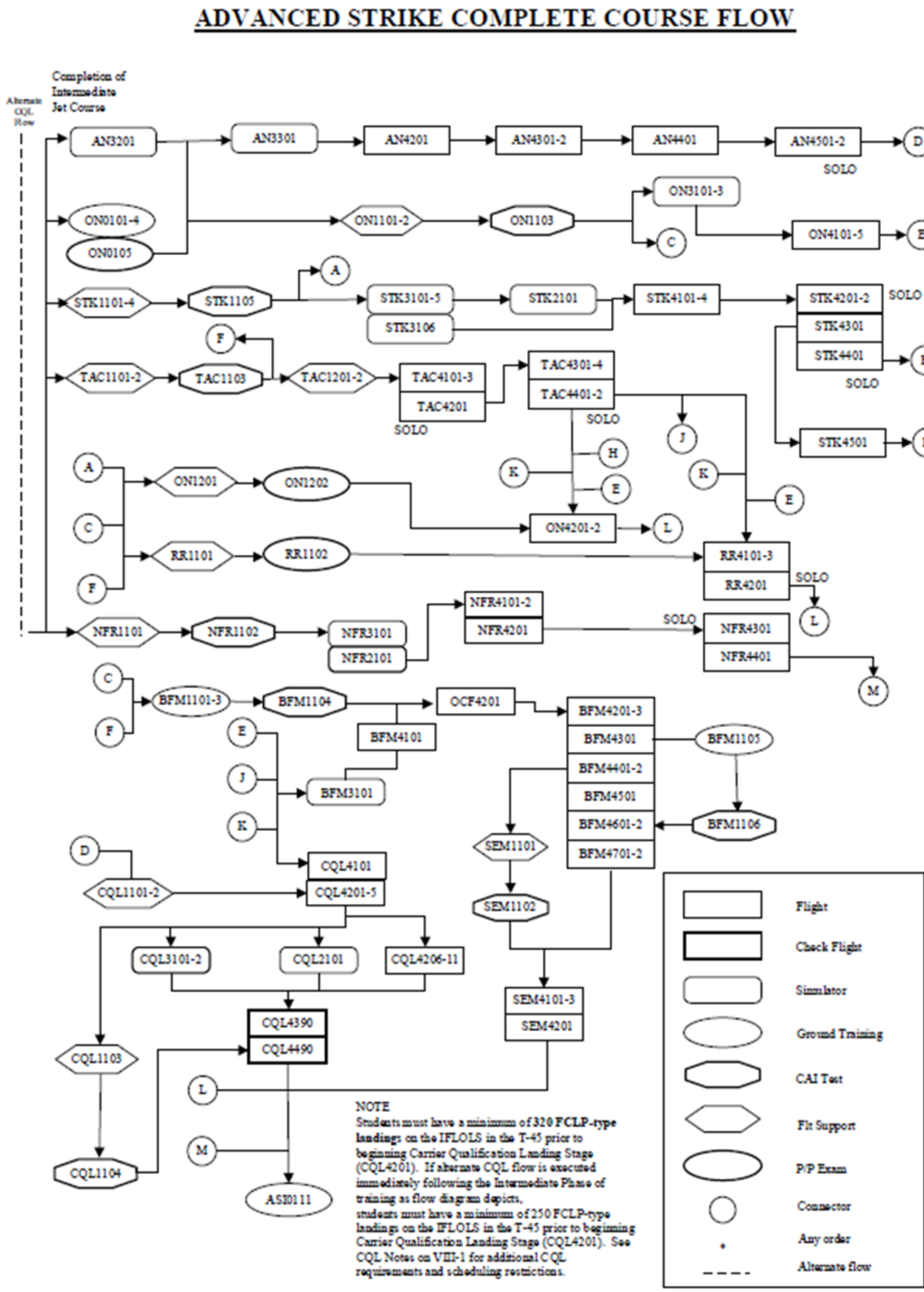

Figure 4. Advanced Strike Complete Flow. Source: Naval Air Training Command (2019).

2. Weather

While TEST considers many scheduling factors, there are three essential factors that TEST does not consider. This first of these is flight weather requirements. Each flight event in the syllabus has a minimum weather requirement. The requirement has two parts, ceiling and visibility. Ceiling refers to the lowest forecasted broken or overcast cloud layer, measured in feet above ground level (AGL) while "visibility is determined through the ability to see and identify preselected and prominent objects at a known distance from the usual point of observation" (U.S. Department of Transportation [DOT] 2019, p. 470). The minimum requirement for both ceiling and visibility must be met in order to fly. In addition to ceiling and visibility, factors such as wind, lightning, and bird migration also influence whether a flight can be executed. Considering weather in scheduling allows for a more robust schedule.

3. Combined Flight Events

Certain pairs of flight events in the syllabus can be combined to save time; the flight briefs for two events are merged into one flight brief. The combined pair of flights is called an in-and-in or an out-and-in flight. The distinction between the two is that in-and-in flights require a landing at the home airport between the two flights; that requirement does not exist for out-and-in flights. In-and-in flights allow for instructors to step out if they are not needed for the second flight. For the purposes of this thesis, we consider only in-and-in flights because all out-and-in flights are encompassed in the list of in-and-in flights. However, we are conservative in our assignment of instructors, assuming that an instructor cannot step out for the second flight and allow VT-22 to manually correct this in the schedule. Utilizing combined flight events reduces scheduled event time.

4. Student Currency

Schedulers must track when each student completed their previous event. This is because each event has a currency requirement. If it has been longer than a certain number of days since the student completed a prerequisite event, they must redo the prerequisite or a similar event to reinstate their currency and be scheduled for the follow-on event. When a student falls out of currency, this is known as falling into a warmup window. The event the student must do to get current is called a warmup event. The currency requirements are displayed in Table 1. For the combinations of event types not listed, there is no currency requirement.

Table 1. Event Currency Requirements

Prerequisite Event Type	Follow-On Event Type	Number of Days Currency Lasts
Lecture	Lecture	14
Lecture	Simulation	14
Lecture	Flight	14
Simulation	Simulation	14
Simulation	Flight	7
Flight	Flight	14

D. CONTRIBUTIONS AND OUTLINE

This thesis extends the development of an aviation scheduling tool. It develops TEST-2, an optimization model designed with the goal of minimizing overall aviation time-to-train. Chapter II provides a literature review of the relevant aviation scheduling literature. Chapter III provides the formulation, assumptions, goals, and limitations of the model, TEST-2. Chapter IV describes how to apply TEST-2 and lists information from a few sample schedules. Chapter V discusses conclusions of this thesis and recommendations for future work.

II. LITERATURE REVIEW

Within the operations research field, researchers often address various scheduling problems. As a result, there exists a copious amount of published research for both military and civilian scheduling. A small subset of this research focuses on military aviation scheduling. In this chapter, we focus on a few specific papers regarding military scheduling.

This thesis directly extends the work of Meditz (2019). Meditz created the Training Event Scheduling Tool (TEST) for VT-22, which is an integer linear program that outputs daily or weekly schedules for students and instructors (Meditz 2019). TEST takes an Excel spreadsheet for its input data and generates a spreadsheet that contains a schedule with hourly resolution (Meditz 2019). The model takes into consideration "desired" events, completed events, non-availability, and the stages of training for each student (Meditz 2019). For each instructor, TEST considers their qualifications and non-availability (Meditz 2019). Additionally, TEST factors in prerequisites, duration of events, and resources such as jet availability (Meditz 2019). Meditz also ensures that TEST serves as a persistent model by setting the completion of "desired" events as a primary objective (Meditz 2019). While Meditz incorporates many constraints into TEST that are unique to military aviation scheduling, TEST does not take into account weather, SNA currency, and combined flight events when building schedules.

A. PREVIOUSLY CITED LITERATURE

Meditz (2019) performs a recent and thorough review of much of the relevant literature for aviation scheduling. A more comprehensive review of the works can be viewed in Chapter II of Meditz (2019).

Jacobs (2014) develops the Flight Training Scheduler (FTS), a tool that creates seven days of schedules for VT-22. FTS does not include many relevant constraints however, such as crew rest and student warm up windows (Jacobs 2014). Additionally, FTS does not include the Intermediate phase of training, nor does it include lectures, night events, and simulator events (Jacobs 2014). Another relevant scheduling tool is

Scheduling Assistance Tool (SAT), created to generate daily flight schedules for NAS Fallon (Slye 2018). SAT schedules events but does not specifically assign students and instructors to these events (Slye 2018). Another limitation of SAT is that it does not consider the training syllabus (Slye 2018).

B. CURRENT LITERATURE

In a student capstone effort from the United States Naval Academy, Frazier, Harris, Loftus, and Pham (2020) use simulation to estimate time-to-train for SNAs at VT-22. While the simulation model does not create a schedule, it does provide an estimate of time-to-train given VT-22's current resource allocation (Frazier et. al 2020). Additionally, the model shows how adjusting certain resources, such as number of available instructors, impacts the time–to-train. This information is relevant for scheduling because we are trying to optimize our schedule, and one of the goals of our objective function is to schedule as many events as possible. We need to know what the limiting factors are for scheduling more events. Frazier et. al found that overall time-to-train decreased with each instructor pilot added, but there was a point of diminishing returns after four instructor pilots added. This is likely due to the fact that another resource such as jets becomes a limiting resource at that point (Frazier et. al 2020). We investigate in our analysis whether adding instructor pilots has a positive impact on the number of events scheduled, which would help with reduction of overall time-to-train.

III. TRAINING EVENT SCHEDULING TOOL VERSION 2.0

This chapter describes the revised Training Event Scheduling Tool (TEST) optimization model, henceforth referred to as Training Event Scheduling Tool Version 2.0 (TEST-2). We discuss its objective, goals, modeling assumptions, formulation, and limitations.

A. OBJECTIVE

The objective of TEST-2 is to build a daily or weekly schedule that assigns each student events in the flight syllabus with the necessary instructors during allowable time periods. TEST-2 does not differ from the original TEST in its primary objective, which is to complete "desired" events. However, TEST-2 considers more factors in the overall objective. TEST-2 continues to penalize "desired" events that are incomplete at the end of the schedule horizon. It also penalizes scheduling during undesirable periods, typically the weekends. TEST-2 penalizes instructor assignments slightly so that that resource is only used when necessary. Additionally, TEST-2 rewards scheduling "possible" events at any time during the week and scheduling "desired" events earlier, as TEST did. TEST-2 differs from TEST in that it rewards schedules that incorporate in-and-in events, which are the combined substitutable events that count as a completion for two flight events. The objective of TEST-2 also incorporates a reward structure for weather considerations. Scheduling events with stricter weather requirements in periods of better weather produces a higher reward than scheduling events with lesser weather requirements. This reward motivates the model to prioritize periods with better forecasted weather for events that require better weather.

B. GOALS

Here, we outline the goals of the original TEST and of TEST-2. The schedulers at VT-22 provided the goals for TEST and TEST-2. Unless otherwise noted, all of the goals from TEST are still considered in TEST-2.

1. TEST Goals

TEST goals are considered in three categories: student, instructor, and overall goals. The main student goals are minimizing time to train, staying current, and maintaining a minimum number of flights each week. Minimizing time to train helps the Navy to maintain force readiness standards. TEST models this goal by penalizing failure to schedule “desired” events, providing a reward for scheduling events in the “possible” list, and rewarding scheduling events early in the week (Meditz 2019). Keeping students current, the second goal, is important because students who are not current must complete a warmup event. A student loses currency when it has been a certain number of days since they completed the prerequisite event for an event. If a student needs a warmup event, they are backtracking in the syllabus rather than progressing through it. This affects our first goal by increasing time to train. Additionally, a warmup event uses resources that could otherwise be used to help a student progress through the syllabus. To model staying current, TEST uses the penalty on “desired” events to achieve this as well. For students who are close to being in a warmup window, the SKEDOs can prioritize completion of certain events using the “desired” event list (Meditz 2019). The final goal of maintaining a minimum number of flights a week is essential because it ensures that each student progresses through the syllabus. Because aircraft and instructors are limited resources, a fair distribution of the resources needs to be considered so each student has a chance to fly every week. Also, in some squadrons, students who are closer to the end of the syllabus can be prioritized over other students, who may then lose currency for their next events in the syllabus. This goal should help mitigate that tendency. To maintain a minimum number of flights, TEST utilizes an elastic constraint that requires each student to fly at least one flight event that week if there are any flights in the student’s “possible” list (Meditz 2019). This constraint only implements a penalty if violated.

The overall goals of TEST are to output easily digestible schedules quickly and to create robust schedules. While creating schedules by hand is time-intensive, TEST is able to create a one-day schedule in minutes. Each scheduled event has time of day, event name, student name, and instructor name.

2. TEST-2 Goals

TEST-2 maintains all goals defined in TEST, improves on some of them, and introduces a few new goals. The goals TEST-2 improves on are student currency, building robust schedules, and minimizing instructor idle time. The new goals are reducing scheduled event time and creating a user-friendly interface. TEST-2 models student currency by incorporating data on completion times for previously executed events. When scheduling events, TEST-2 considers the currency requirement by comparing when the student last completed a prerequisite event, and if necessary, schedules a warmup event for the student to complete. To incentivize scheduling warmups so students become current instead of simply passing over them, we built in a reward for completing warmup events. For improving on the robustness of schedules, TEST-2 considers weather requirements for each event and takes as an input the weather forecast broken down by period. The reward structure prioritizes good weather periods for events with stricter weather requirements. Additionally, events are only scheduled if the weather requirements are met for the entire duration of the event, not just for the takeoff period. These weather considerations help augment the robustness of TEST-2's schedules. For reduction of scheduled event time, TEST-2 introduces combined flight events. When two flight events are combined, their briefings are also combined. The combined briefings reduce scheduled event time by about one hour.

Because TEST-2 is not integrated with Training Command Sierra Hotel Aviation Readiness Program (TSHARP), we need a different medium for TEST-2 to host the model. Eckstrand (2020) develops an interface to host TEST-2. This interface allows a user to upload an input data spreadsheet, press solve, then wait for TEST-2 to create the schedule. Once TEST-2 builds the schedule, the interface then displays it on the same page. A view of the interface is displayed in Figure 5.

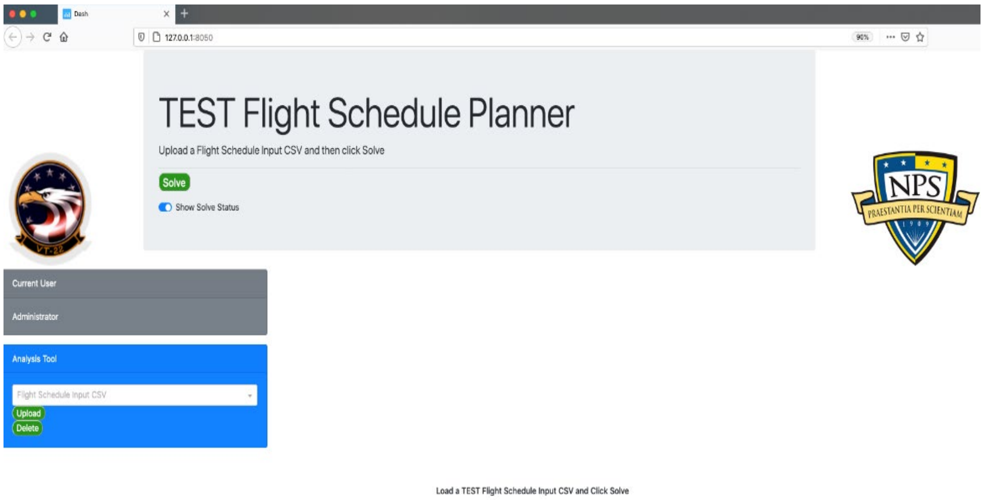

Figure 5. TEST-2 Interface. Source: Eckstrand (2020).

C. ASSUMPTIONS

1. TEST Assumptions

In the original TEST formulation, Meditz made four main assumptions. The first is that using an hourly resolution for scheduling is adequate, providing enough detail while allowing for good computation times. The second is that aircraft maintenance has no effect on scheduling. The third assumption is that crew rest is always from the end of one day to the start of the next day. The last assumption is that if a student has any flight events in their "possible list," they should be scheduled for at least one flight event that week (Meditz 2019).

2. TEST-2 Assumptions

TEST-2 maintains the original assumptions created for TEST while adding some additional assumptions. We assume that if possible, it is always preferable to do an in-and-in or out-and-in event instead of two events separately. We also assume that the weather forecast is only relevant for the local airport. Also, we do not consider the weather during the brief or debrief periods of a flight event. For combined flight events, we assume for simplicity that in-and-in and out-and-in events are interchangeable and only model in-and-in events. Another assumption of TEST-2 is that flight events cannot be scheduled before 0800 or after 2345 because of field hours. Additionally, we assume that civilian instructors are readily available and do not consider their availability in our model for scheduling lecture or simulator events, which they instruct.

D. FORMULATION

TEST-2 is an integer linear program. Its objective function and constraints motivate the model to generate solutions that achieve our goals. The necessary data, sets, decision variables, and constraints are outlined below. Much of the original formulation of TEST is unchanged for TEST-2. Unless otherwise noted by red text, the formulation below is from Meditz (2019).

1. Indices [~cardinality]

D	The set of days: $d \in D = \{D1, D2, ...D7\}$ (ordered set) [7].
E	The set of all student events: $e \in E = \{CO3101, \ldots\}$ [258].
I	The set of instructors: $i \in I$ [38].
J	The set of stages: $j \in J = \{CO, EP, ...E2/C2\_CQ\}$ [24].
P	The set of periods: $p \in P = \{p1, p2, ...p168\}$ (ordered set) [168].
S	The set of students: $s \in S$ [73].

2. Subsets[1]

$A_e \subset P$	Periods where event e is allowed to start.
$DD \subset E \times E$	$(e', e) \in DD$ if event e' cannot be the same day as event e.
$E^f \subset E$	Events that are flight events.
$E^j \subset E$	Events that are in stage j.
$E^{lec} \subset E$	Events that are lectures.

[1] We identify new model elements that are not in Meditz (2019) using red text.

$E^{only} \subset E$	Events that can be scheduled only with lecture events occurring the same day.
$E^{onlyFlight} \subset E$	Flight events that can be scheduled only with lectures and simulator events occurring the same day.
$E^{NOW} \subset E$	Events that should not be flown with on-wing instructor.
$E^{OW} \subset E$	Events required to be flown with on-wing instructor.
$E^{sim} \subset E$	Events that are simulator events.
$E^{II} \subset E$	Events that are in-and-in events.
$I_e \subset I$	Instructors who can instruct event e.
$OW_s \subset I$	Instructors who are on-wing for student s.
$P_d \subset P$	Periods in day d (ordered subset).
$P_i^{NO} \subset P$	Periods where instructor i is not available.
$P_s^{NO} \subset P$	Periods where student s is not available.
$R \subset E \times E$	$(e', e) \in R$ if event e' precedes event e.
$S_e \subset E$	Events e' that satisfy event e.
$SP_{ep} \subset P$	All periods p' that prohibit starting any event if event e started in period p.
$warmups \subset E \times E$	$(e', e) \in warmups$ if event e' reinstates currency for event e.
$warmupS_e \subset E$	Events e' that satisfy warmup event e.

3. Data[2]

[2] We identify new model elements that are not in Meditz (2019) using red text.

$Buffer_e$	Time required after event e before starting the next event. [periods]
CAP	Maximum number of students that can be assigned to a lecture event in a given period. [students] Default $CAP = 20$.
$complete_{e,s}$	1 if student s already completed event e, 0 otherwise.
$CQwave_p$	Maximum number of students that can be waved during a CQ event in period p. [students]
$currency_{e'e}$	Pair of days that indicates that maximum number of days until a student is in a warmup window for either 1 warmup or 2 warmups, necessitating they redo event e' either once or twice before being able to schedule event e. [number]
$days_{e,s}$	Number of days since student s completed event e at the beginning of the week. [number]
$EEvent_{d,i}$	1 if instructor i can be scheduled for an extra event on day d. [events]
$forecast_p$	Pair of values (ceiling, visibility) that are the expected weather forecast during period p. [number]
$G_{e,s}$	1 if requiring completion of event e for student s (desired), 0 otherwise.
$ii\,\mathrm{R}_e$	Reward for scheduling an in-and-in event, e. [number]
$instructorP$	Penalty for scheduling instructors to events. [number] Default $instructorP = 0.01$
M_p	Maximum number of flights in period p (jet availability). [aircraft]

Default $M_p = 21 \; \forall p \in P$.

NA_e Number of aircraft required for event *e*. [aircraft]

NI_e Number of instructors required for event *e*. [instructors]

NUM_e Number of periods in event *e*. [periods]

$numwarmups_{e,p,s}$ Number of warmups (0, 1, or 2) necessary for student *s* to complete event *e* in period *p*. If a student is current, 0, otherwise, 1 or 2 depending on the event. [number]

$possible_{e,s}$ 1 if student *s* can complete event *e* during the week, 0 otherwise. Calculated to be all events in $stage_{j,s}$ that are not in $complete_{e,s}$ where $stage_{j,s}$ is 1 if student s is in stage *j* and 0 otherwise. Also 1 if *e* is a warmup event a student needs to complete to be current to conduct other possible events, even if student *s* has previously completed *e*.

$Reward_{e,p,s}$ Reward for scheduling student *s* to event *e* in period *p*. [number] Calculated as $scheduleR_{e,p,s} exR_p$ $\forall e \in E, p \in P, s \in S$ where $scheduleR_{e,p,s}$ is the reward for scheduling student *s* to event *e* in period *p* [number] and exR_p is the reward for scheduling an event in period *p* that is earlier in the week, which is calculated as 0.99^{p-1}. [number]

$scheduleP_p$ Penalty for scheduling an event in period *p*. [number]

$simCAP$ Maximum number of students that can be in a simulator event in any period. [students]

Default $simCAP = 5$.

$studentP_{e,s}$ Penalty for student *s* not completing event *e*. [number]

$warmupR_{e,p,s}$ — Reward for student s completing warmup event e in period p. [number]

$weather_e$ — Pair of values (ceiling, visibility) that are the minimum weather required to schedule flight event e. [number]

$weatherR_{e,p}$ — Reward for scheduling event e in period p. Calculated using the lookup table shown below. [number]

Table 2. Weather Reward Lookup Table

		Weather Forecast			
		VFR	**MVFR**	**IFR**	**LIFR**
Weather Requirement	**VFR**	4	-	-	-
	MVFR	3	2.5	-	-
	IFR	2	1.5	-	-
	LIFR	1	0.5	0.25	0.1

4. Decision Variables

$ELAS_s$ — Non-negative variable with value of 1 if student s is not scheduled to a flight event in a week.

$L_{e,i,p}$ — Binary variable with value of 1 if instructor i instructs lecture event e in period p and 0 otherwise.

$LEC_{d,s}$ — Binary variable with value of 1 if student s has more than four lecture hours on day d and 0 otherwise.

$LECI_{d,i}$ — Binary variable with value of 1 if instructor i instructs more than four lecture hours on day d and 0 otherwise.

$Y_{e,i,p}$ — Binary variable with value of 1 if instructor i flies event e in period p and 0 otherwise.

$X_{e,p,s}$ — Binary variable with value of 1 if student s starts event e in period p and 0 otherwise.

5. Objective[3]

$$
\begin{aligned}
&Min \sum_{e|G_{e,s}=1,s} (G_{e,s} studentP_{e,s} - \sum_{p \in A_e} X_{e,p,s}) + \sum_{e,p \in A_e,s} scheduleP_p X_{e,p,s} \\
&+ \sum_{e|possible_{e,s}=1, p \in A_e, s} Reward_{e,p,s} X_{e,p,s} + \sum_{e,i,p \in A_e} instructorP(Y_{e,i,p} + L_{e,i,p}) \\
&+ \sum_{s} flightP(ELAS_s) + \sum_{e,p \in A_e,s} iiR_e X_{e,p,s} + \sum_{e,p \in A,s} warmupR_{e,p,s} X_{e,p,s} \\
&+ \sum_{e,p \in A,s} weatherR_{e,p} X_{e,p,s}
\end{aligned}
\tag{0}
$$

6. Constraints

$$
X_{e,p,s} \le \sum_{\substack{p' \le p - NUM_{e''} - Buffer_{e''} \cap p' \in A_{e''} \\ e'' \in S_{e'} | possible_{e'',s}=1}} X_{e'',p',s} \qquad \begin{aligned} &\forall (e',e) \in R \,| \\ &possible_{e,s}=1, \\ &complete_{e',s}=0, \\ &possible_{e',s}=1, \\ &p \in A_e, s \in S \end{aligned} \tag{1}
$$

$$
X_{e,p,s} \le \frac{1}{numwarmups_{e,p,s}} \sum_{\substack{p' \le p - NUM_{e''} - Buffer_{e''} \cap p' \in A_{e''} \\ e'' \in S_{e'} | possible_{e',s}=1}} X_{e'',p',s} \qquad \begin{aligned} &\forall (e',e) \in warmups \,| \\ &possible_{e,s}=1, \\ &complete_{e',s}=1, \\ &possible_{e',s}=1, \\ &numwarmups_{e,p,s} > 0, \\ &p \in A_e, s \in S \end{aligned} \tag{2}
$$

$$
\sum_{\substack{p \in A_{e'} \\ e' \in S_e}} X_{e',p,s} \le 1 \qquad \begin{aligned} &\forall e \in E \,|\, possible_{e,s}=1, \\ &s \in S \end{aligned} \tag{3}
$$

$$
\sum_{\substack{p \in A_{e''} \\ e'' \in warmupS_{e'}}} X_{e'',p,s} \le numwarmups_{e,p,s} \qquad \begin{aligned} &\forall (e',e) \in warmups \,| \\ &possible_{e,s}=1, \\ &numwarmups_{e,p,s} > 0, \\ &s \in S \end{aligned} \tag{4}
$$

$$
\sum_{e|possible_{e,s}=1, p - NUM_e - Buffer_e + 1 \le p' \le p \cap p' \in A_e} X_{e,p',s} \le 1 \qquad \forall p \in P, s \in S \tag{5}
$$

[3] We identify new model elements that are not in Meditz (2019) using red text.

$$\sum_{e,p-NUM_e-Buffer_e+1\le p'\le p\cap p'\in A_e}(Y_{e,i,p'}+L_{e,i,p'})\le 1 \qquad \forall i\in I, p\in P \qquad (6)$$

$$\sum_{e\in E^f|possible_{e,s}=1,p\in A_e} X_{e,p,s}\ge 1-ELAS_s \qquad \forall s\in S \mid \sum_{e\in E^f} possible_{e,s}\ge 1 \qquad (7)$$

$$\sum_{e\in(E^{sim}\cup E^f)\cap e\notin(E^{CQ}\cup E^{only})|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 2(1-LEC_{d,s}-\sum_{e\in E^{only}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}) \qquad \forall d\in D, s\in S \qquad (8)$$

$$\sum_{e\in E^f\cap e\notin(E^{CQ}\cup E^{onlyFlight})|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 2(1-\sum_{e\in E^{onlyFlight}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}) \qquad \forall d\in D, s\in S \qquad (9)$$

$$\sum_{e\in E^{lec}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}NUM_e\le 4+4LEC_{d,s} \qquad \forall d\in D, s\in S \qquad (10)$$

$$\sum_{e\in E^{onlyFlight}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 1-LEC_{d,s} \qquad \forall d\in D, s\in S \qquad (11)$$

$$\sum_{e\in E^{lec}\cap e\notin E^{only}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 8(1-\sum_{e\in E^{only}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}) \qquad \forall d\in D, s\in S \qquad (12)$$

$$\sum_{e\in E^{lec}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 8(2-\sum_{e\in(E^f\cup E^{sim})\cap e\notin E^{CQ}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}) \qquad \forall d\in D, s\in S \qquad (13)$$

$$\sum_{e\in E^{lec}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 8(3-\sum_{e\in E^{CQ}\cap e\notin E^{lec}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}) \qquad \forall d\in D, s\in S \qquad (14)$$

$$\sum_{e\in E^{CQ}\cap e\notin E^{lec}|possible_{e,s}=1,p\in P_d\cap A_e} X_{e,p,s}\le 3(1-LEC_{d,s}) \qquad \forall d\in D, s\in S \qquad (15)$$

$$\sum_{e'|possible_{e',s}=1,p'\in(SP_{e'p}\cap A_{e'})} X_{e',p',s}\le 8(1-\sum_{e|possible_{e,s}=1} X_{e,p,s}) \qquad \forall s\in S, p\in P \qquad (16)$$

$$\sum_{e \in E^f, p \in P_d \cap A_e} Y_{e,i,p} \leq (2 + EEvent_{d,i})(1 - LECI_{d,i}) \qquad \forall d \in D, i \in I \qquad (17)$$

$$\sum_{e \in E^{lec}, p \in P_d \cap A_e} L_{e,i,p} NUM_e \leq 4 + 4LECI_{d,i} \qquad \forall d \in D, i \in I \qquad (18)$$

$$\sum_{e', p' \in (SP_{e'p} \cap A_{e'})} (Y_{e',i,p'} + L_{e',i,p'}) \leq 8(1 - \sum_{e} (Y_{e,i,p} + L_{e,i,p}) \qquad \forall i \in I, p \in P \qquad (19)$$

$$\sum_{s | possible_{e,s} = 1} X_{e,p,s} \leq \sum_{i \in I_e} L_{e,i,p} CAP \qquad \forall e \in E^{lec}, p \in A_e \qquad (20)$$

$$\sum_{e \in E^{CQ} \cap E^f, s | possible_{e,s} = 1} X_{e,p,s} \leq CQwave_p \qquad p \in A_e \mid CQwave_p > 0 \qquad (21)$$

$$\sum_{e | possible_{e,s} = 1, p - NUM_e + 1 \leq p' \leq p | p' \in A_e, s} NA_e X_{e,p',s} \leq M_p \qquad \forall p \in P \qquad (22)$$

$$\sum_{s | possible_{e,s} = 1} NI_e X_{e,p,s} = \sum_{i \in I_e} Y_{e,i,p} \qquad \forall e \in E^f, p \in A_e \qquad (23)$$

$$\sum_{p \in P_d \cap A_e} X_{e,p,s} + \sum_{p \in P_d \cap A_{e'}} X_{e',p,s} \leq 1 \qquad \begin{array}{l} \forall d \in D, (e', e) \in DD \\ \mid (possible_{e,s} = 1, \\ possible_{e',s} = 1, \\ complete_{e',s} = 0, \\ complete_{e,s} = 0), \\ s \in S \end{array} \qquad (24)$$

$$X_{e,p,s} \leq \sum_{i \in I_e \cap OW_s} Y_{e,i,p} \qquad \begin{array}{l} \forall e \in E^{OW}, p \in A_e, \\ s \in S \mid possible_{e,s} = 1 \end{array} \qquad (25)$$

$$X_{e,p,s} \leq \sum_{i \in I_e | i \notin OW_s} Y_{e,i,p} \qquad \begin{array}{l} \forall e \in E^{NOW}, p \in A_e, \\ s \in S \mid possible_{e,s} = 1 \end{array} \qquad (26)$$

$$\sum_{e \in E^{OW} | p \in A_e, s | possible_{e,s} = 1 \cap i \in (I_e \cap OW_s)} X_{e,p,s} \leq 1 \qquad \forall i \in I, p \in P \qquad (27)$$

$$\sum_{s, e \in E^{sim} | possible_{e,s} = 1, p - NUM_e + 1 \leq p' \leq p | p' \in A_e} X_{e,p',s} \leq simCAP \qquad \forall p \in P \qquad (28)$$

$$\sum_{p-NUM_e+1 \le p' \le p \cap p' \in A_e} (Y_{e,i,p'} + L_{e,i,p'}) = 0 \qquad \forall e \in E, i \in I, p \in P_i^{NO} \qquad (29)$$

$$\sum_{p-NUM_e+1 \le p' \le p \cap p' \in A_e} X_{e,p',s} = 0 \qquad \forall e \in E, p \in P_s^{NO}, s \in S \qquad (30)$$

$$Y_{e,i,p}, L_{e,i,p} \in \{0,1\} \qquad \forall e \in E, i \in I, p \in P \qquad (31)$$

$$X_{e,p,s} \in \{0,1\} \qquad \forall e \in E, p \in P, s \in S \qquad (32)$$

$$LEC_{d,s} \in \{0,1\} \qquad \forall d \in D, s \in S \qquad (33)$$

$$LECI_{d,i} \in \{0,1\} \qquad \forall d \in D, i \in I \qquad (34)$$

$$ELAS_s \ge 0 \qquad \forall s \in S \qquad (35)$$

The objective function (0) expresses the cost of not scheduling events in the "desired" list, the penalties for scheduling in undesirable periods, assigning instructors to events, and not scheduling students to at least one flight event. It also includes the rewards for completing additional events, scheduling in-and-in events, scheduling warmup events to bring students to currency, and allowing events with stricter weather requirements to utilize better forecasted weather.

Constraint (1) ensures adherence to precedence relationships between events in the syllabus.

Constraint (2) ensures student currency guidelines are followed.

Constraint (3) ensures each event is only completed at most once by each student. If an event has multiple substitutable events that satisfy completion, the constraint only allows one of the events to be completed by the student.

Constraint (4) ensures all warmup events are completed no more than necessary, either once or twice depending on the currency requirement. If an event has multiple substitutable warmup events that reinstate currency, the constraint only allows up completion of one or two of these events, total, based on the currency requirement.

Constraint (5) ensures each student is scheduled for at most one event in each period.

Constraint (6) ensures each instructor instructs at most one flight or lecture event in each period.

Constraint (7) ensures each student is scheduled for at least one flight event in the week if there is a flight event in his or her "possible" list or records a deviation from this requirement.

Constraint (8) requires that students complete no more than two flight or simulator events per day except for CQ events and events that should be only scheduled one per day.

Constraint (9) requires that students complete no more than two flight events per day except for when a student has a flight event that should not be completed with any other flight event.

Constraint (10) allows students to complete up to four lecture hours on days where they have a simulator or flight event and up to eight lecture hours if they do not have a simulator or flight event.

Constraint (11) ensures flight events that should only be completed one per day are not paired with more than four hours of lectures on the same day.

Constraint (12) ensures events that should only be completed one per day are not paired with any lecture events.

Constraint (13) ensures that if a student has two flight or simulator events per day, he or she should not be scheduled for any lecture events that day.

Constraint (14) ensures that if a student has three flight or simulator events per day while in the CQ stage, he or she should not be scheduled for any lecture events that day.

Constraint (15) allows students to complete up to three CQ flight or simulator events per day if they do not have more than four lecture hours on the same day.

Constraint (16) ensures adherence to student crew day limitations.

Constraint (17) requires that instructors teach no more than two flight or simulator events per day unless they have a "surge" event.

Constraint (18) allows instructors to teach up to four lecture hours on days where they fly and up to eight lecture hours if they do not have a flight event.

Constraint (19) ensures adherence to instructor crew day limitations.

Constraint (20) ensures instructors are scheduled to instruct lecture events while adhering to classroom capacities.

Constraint (21) allows LSOs to wave up to a certain number of students in a given period.

Constraint (22) ensures adherence to jet availability.

Constraint (23) ensures the correct number of instructors are assigned to fly each event.

Constraint (24) ensures certain events are scheduled on different days.

Constraint (25) requires students fly with their on-wing for on-wing events.

Constraint (26) requires students do not fly with their on-wing for non on-wing events.

Constraint (27) ensures that if multiple students share an on-wing instructor, they do not start the on-wing event in the same period.

Constraint (28) restricts the number of students assigned to simulator events in a given period.

Constraint (29) and (30) adheres to instructor and student non-availabilities.

Constraints (31) to (35) declare variable types.

E. LIMITATIONS

Meditz identified three main limitations of TEST. First, TEST is not integrated with VT-22's current management and scheduling system, TSHARP. Second, TEST requires the user to manually input data. Third, TEST does not distinguish between lead instructors and other instructors for multiplane events (Meditz 2019). All three of these limitations still exist for TEST-2.

IV. USING TEST-2 AND ANALYSIS

A. DATA COLLECTION

VT-22 collected data over the week of November 2, 2020 to November 6, 2020. This data serves as the input data for TEST-2 for the purposes of our analysis. The data consists of instructor and student non-availabilities, instructor qualifications, jet availability, student currency information, the weather for the week, and student progress through the syllabus. Additionally, VT-22 provided us with both the schedules they created and the actual schedule completed for each day; these differ due to flight cancellations. The main factors that led to flight cancellations were bad weather, jet non-availability, and instructor non-availability. For this week, VT-22 had 73 students and 38 available instructors. The schedulers planned for 394 events, and 345 events were actually completed.

B. MODEL INPUTS

From the data VT-22 provided, we created an Excel spreadsheet with multiple sheets to serve as the model input data. We used the same initial format as Meditz (2019) but added additional sheets to contain the additional data required for TEST-2. A majority of the sheets in the input spreadsheet remain constant between VT-22's planning cycles, but a few of them require manual updates daily or weekly from the schedulers. For example, the schedulers must provide the list of "desired" events for each student and must update each student's current flight stage(s). The schedulers also must update the input data with student completions. If new students or instructors arrive or leave the squadron, schedulers must manually add or remove them.

1. TEST Inputs

Among the input data sheets, some sheets remain unchanged in structure from Meditz (2019). For these sheets, we simply provided updated data. These sheets include "desired" events for each student, student completed events, which instructors and students are currently at the squadron, jet availability, instructor and student non-availabilities, instructor qualifications, student flight stages, and student on-wing instructors. Input sheets that did not require any updates from Meditz (2019) are the prerequisite matrix, list of events, length of events, and model penalties.

As in Meditz (2019), the SKEDSO must manually enter the "desired" events for each student, which are the events they wish the student to complete that week. An example of how to enter "desired" events for a few students is shown in Table 3. An entry of "1" indicates the event is desired to be completed for that student and an empty cell indicates the event is not a "desired" event.

Table 3. Student "Desired" Event Example Input

Student	RI1101-05	RI1106	RI3101	RI3102	RI3103	RI3104	RI3201	RI3102
ENS A	1	1	1	1				
1stLt B			1	1	1	1		
LTJG C						1	1	1

Additionally, the SKEDSO needs to input which events the student has completed at the end of each day. An example of how to enter completed events for a few students in shown in Table 4. An entry of "1" indicates that the event has been completed for that student and an empty cell indicates that event has not yet been completed by the student.

Table 4. Student Event Completion Example Input

Student	RI1101-05	RI1106	RI3101	RI3102	RI3103	RI3104	RI3201	RI3102
ENS A								
1stLt B	1	1						
LTJG C	1	1	1	1	1			

Currently, the squadron tracks personnel availability in TSHARP which outputs a daily Excel sheet that contains non-availabilities due to leave, medical appointments, watch, and other factors. The SKEDSO enters these instructor and student non-availabilities into two inputs sheets that follow a similar format. These must be updated daily. An entry of "1" indicates that the instructor or student cannot be assigned any event for that period. An empty cell indicates that the instructor or student can be scheduled during that period. An example of how to enter these non-availabilities is shown in Table 5.

Table 5. Instructor Non-Availabilities Example Input

Instructor	**P8 (0800)**	**P9 (0900)**	**P10 (1000)**	**P11 (1100)**	**P12 (1200)**	**P13 (1300)**	**P14 (1400)**	**P15 (1500)**
LtCol X								
LCDR Y		1	1	1	1	1		
LT Z	1	1						

Another input to TEST-2 is instructor qualifications. These are tracked and updated in the Flight Instructor Standardization and Training (FIST) matrix (Figure 6).

VT-22 Designations Report

Created 10/26/2020 09:24:26

Expired | < 15 Days | < 30 Days | Designated | Pending

	INST	FAM	FRM	DIV	NFM	NFMCHASE	NFR	Section Lead	Division Lead	ON	SLL	RR	STK	STK POP	STK LD	OCF	TAC	TAC LD	BFM	BFM LD	SEM	SEM LD	CQ DEMO	CQLDSF	FCF	Instrument Checker	NATOPS	CRM	T-45C IP
LT A	S	S	S	S	S		S	D	D	S	S	S	S	D	D	X	S	D	X	D	X					D		D	
Capt. B	S	S	X	X	X		X	D																		D		D	
LCDR C	X	X	X	X	X		X	D	D	X			X	D	D		X	I	I				D	D	D				
LT D	X	X	X	X	X		X	D	I														D						
Maj. E	S	S	S	S	S		S	D	D	S	S	S	S	D	D	X	S	D	X	D	I								
LtCol F	S	S	S	S	S	X	S	D	D																		S		
LT G	X	X	X	X	X		X	D		X			X	I			Q	I											
LT H	S	S	S	S	S		S	D	D	X	X	I	X			X	S	D	S	D	S	D							
LCDR I	X	X	X	X	X		X	I															D						
LT J	S	S	S	S	S		S	D	D	S	S	S	S	D	D	S	S	D	S	D	S	D				D			
Capt. K	X	X	X	X	X	X	X	D	D	I			X			X	I												
LT L	X	X	X	X	X		X	D	D								X	I					D	I					
LT M	X	X	Q		X																								
LT N	S	S	S	S	S		S	D	D	X	X	X	S	D	D	X	S	D	I									S	
LT O	S	X	Q		X																								

Figure 6. Example FIST Matrix. Adapted from Simpson (2020).

The SKEDSO must transfer the information from the FIST into the TEST-2 input sheet for instructor qualifications. The input format for each event is a list of instructors who are qualified to instruct that event. An example of the input format is displayed in Table 6.

Table 6. Instructor Qualifications Example Input

TAC4201	**TAC4301**	**TAC4302**	**AN3201**	**AN4201**	**AN3301**	**AN4301**
LtCol X	LtCol X	LtCol X		LtCol X		LtCol X
LCDR Y	LCDR Y	LCDR Y		LCDR Y		LCDR Y
	LT Z	LT Z		LT Z		LT Z

The schedulers at VT-22 currently track what stage each student is in on a large whiteboard, seen in Figure 7. Each student's "possible" events to be scheduled for depend on what stage(s) they are currently in.

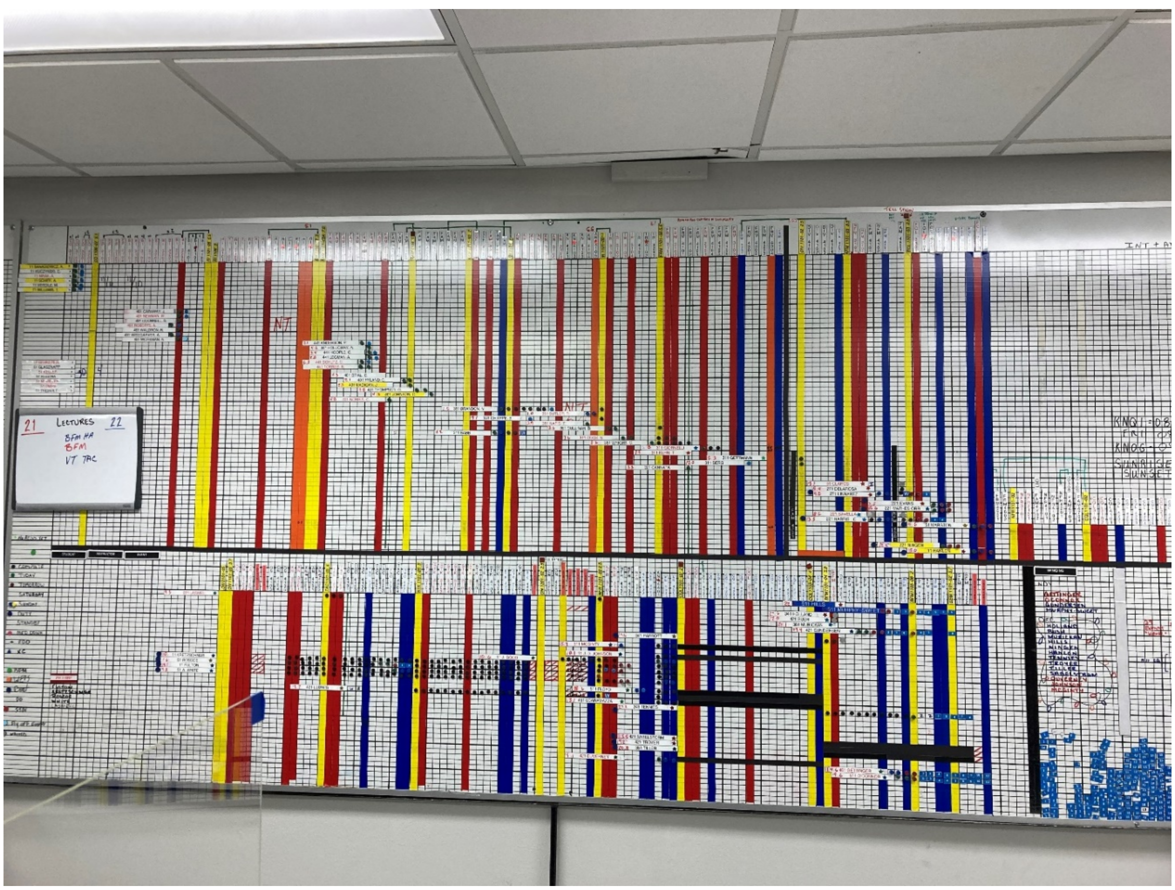

Figure 7. VT-22 Student Progress Whiteboard. Source: Simpson (2020).

In the TEST-2 input sheet, the SKEDSO should enter a "1" for each stage a student is currently in or will be in for the week. We provide an example of input for student stages in Table 7.

Table 7. Student Stage Example Input

Student	CO	EP	BI	RI	FAM/OCF	AN	IR	FRM
ENS A			1	1				
1stLt B				1				
LTJG C				1				

2. TEST-2 Inputs

We added model input sheets regarding the weather forecast for the week, weather requirements, combined flight events, student currency requirements, and student event completion dates. Weather forecast and student event completion dates require daily changes. Weather requirements, combined flight events, and student currency requirements would be updated less frequently.

The weather forecast is broken down by each period and consists of both a ceiling and visibility estimate. The SKEDSO must manually update this forecast each time they run TEST-2. A sample weather forecast is shown in Table 8.

Table 8. Sample Weather Forecast

Period	Ceiling (feet above ground level)	Visibility (statute miles)
P8 (0800)	3000	5
P9 (0900)	3000	5
P10 (1000)	2000	3
P11 (1100)	1000	2
P12 (1200)	1000	1
P13 (1300)	2000	3
P14 (1400)	2000	3

The schedulers need to update student event completion dates in the same way they update student event completions, except instead of using a “1” to indicate completion, a date should be entered. An empty cell indicates the event has not yet been completed. An example of how to enter completion dates is shown in Table 9.

Table 9. Student Event Completion Dates Example Input

Student	RI1101-05	RI1106	RI3101	RI3102	RI3103	RI3104	RI3201
ENS A							
1stLt B	10/28/2020	10/29/2020					
LTJG C	10/21/2020	10/22/2020	10/23/2020	10/26/2020	10/27/2020		

The student currency requirements sheet should not require regular updating, but we provide an explanation and example of how the input sheet works because it is new for TEST-2. For each event, the schedulers must check to see when the most direct prerequisite event was last completed. If it has been a certain number of days since prerequisite event completion, one or two warmup events must be completed. For each event, we list the prerequisite event to be considered, the event itself, what warmup event needs to be conducted to reinstate currency, the number of days currency lasts until one warmup is needed, and the number of days currency lasts until two warmups are needed. A value of 1000 days for two warmups indicates that only one warmup is required. There is no warmup requirement moving between stages.

Table 10. Student Currency Inputs

Prerequisite	Event	Warmup	Days One	Days Two
BI3204	BI3205	BI3204	14	1000
BI3205	BI4101	BI3205	7	14
BI4101	BI4102	BI4101	14	1000
BI4102	BI4103	BI4102	14	1000
BI3205	RI1101-05	none	N/A	N/A
RI1101-05	RI1106	RI1101-05	14	1000
RI1101	RI3101	RI1106	14	1000

C. MODEL IMPLEMENTATION

We implement TEST-2 using Python version 3.7 and the Pyomo version 5.7 package (Hart et. al 2017). CPLEX Interactive Optimizer version 12.10 solves TEST-2 (IBM 2018). We run TEST-2 on a 64-bit Dell Latitude E5450 with two 2.60 GHz processors and 8 GB of RAM. For the daily and weekly model, we use a relative optimality gap of 1%. We also impose a time limit on the weekly model of four hours, so the model solves until it has a gap of less than 1% or 4 hours is reached, whichever comes first. With these settings, the daily model has about 68,000 constraints and 416,000 binary variables while the weekly instance consists of 417,000 constraints and 3,379,000 binary variables. The daily model solves in less than ten minutes with a 1% optimality gap. The weekly model typically runs until the 4-hour cutoff time and does not reach the 1% optimality gap. For purposes of the main weekly comparison provided in Section D, TEST-2 produces a solution with a 33% gap.

D. MODEL OUTPUT

TEST-2 outputs the created schedule to a CSV file that can be opened in Excel. The schedule for each day first has student names in the leftmost column, then instructors below them. To the right of each student or instructor are their scheduled event(s) and the time periods in which they are scheduled. A sample output is displayed in Table 11.

Table 11. TEST-2 Sample Schedule

Day 1	**P8 (0800)**	**P9 (0900)**	**P10 (1000)**	**P11 (1100)**	**P12 (1200)**
ENS A	F: FRM4102	F: FRM4102	F: FRM4102	F: FRM4102	F: FRM4102
1stLt B	F: FAM4103	F: FAM4103	F: FAM4103	F: FAM4103	
LTJG C			S: IR3103	S: IR3103	S: IR3103
LtCol X	F: FAM4103	F: FAM4103	F: FAM4103	F: FAM4103	
LCDR Y	F: FRM4102	F: FRM4102	F: FRM4102	F: FRM4102	F: FRM4102
LT Z					

Each schedule generated by TEST-2 contains 24 1-hour periods per day. Lecture events are preceded by an “L” while simulator events are preceded by an “S” and flight events an "F." TEST-2 does not assign instructors to lecture or simulator events because those are entirely handled by civilian instructors, and TEST-2 only considers military instructors.

E. NUMERIC COMPARISON

For our test case, TEST-2 generates a weekly schedule containing 68 lectures events, 132 simulator events, and 204 flight events, for a total of 404 events. Of the flight events, 20 are in-and-in flights. The manual schedules VT-22 created for the week included 382 total events. Of those 382 events, only 345 were completed. Of these 345 completed events, 174 were flight events. The cancelled flights were due to three main reasons: lack of jet availability, weather, and instructor non-availabilities. A comparison of the number of events actually completed for the week vs. TEST-2's scheduled events is displayed in Table 12.

Table 12. Number of Events in Manual Schedule vs. TEST-2 Schedule

Event Type	Manual Schedule – Completed	TEST-2 Schedule
Lecture	65	68
Simulator	106	132
Flight	174	204
TOTAL	345	404

We see that TEST-2 is able to schedule more lecture, simulator, and flight events than were actually completed by VT-22 for the week of November 2–6. Because TEST-2 takes jet availability, weather, and instructor non-availabilities into consideration while building the schedule, cancellations due to these factors should be minimal.

F. ADDITIONAL JETS COMPARISON

Jets can be a limiting resource for scheduling at some installations. To test the impact of expanding VT-22's jet fleet, we ran TEST-2 with five additional available jets each period. TEST-2 schedules the same number of total events regardless of the number of jets available. With additional jets available, TEST-2 does not schedule more flights.

Therefore, jets are likely not currently a limiting resource for scheduling at VT-22 for the week considered.

G. ADDITIONAL INSTRUCTORS COMPARISON

VT-22 currently considers instructors to be its most limiting resource in building schedules. We explored the effects of adding both one "super instructor" and five additional qualified instructors on the schedule. The "super instructor" is an instructor qualified to instructor all events whereas the five instructors are not qualified to instruct all events. We chose to explore the effects of adding up five instructors because Frazier et. al found a reduction in time-to-train with up to five instructors.

After adding the one "super instructor," the schedule built by TEST-2 does not schedule more flight events overall than the original schedule. However, TEST-2 schedules the "super instructor" for eight flight events over the course of the week, most of which are events that have less than ten qualified instructors. An instructor with more qualifications allows for more flexibility in scheduling.

With the five added instructors, TEST-2 generated a schedule with 412 total events, which consists of 69 lecture events, 133 simulator events, and 210 flight events. We provide a comparison of the number of events TEST-2 schedules with its current 38 instructors vs. 43 instructors in Table 14.

Table 13. Number of Events with and without Additional Instructors

Event Type	TEST-2 Schedule	TEST-2 Schedule with Additional Instructors
Lecture	68	69
Simulator	132	133
Flight	204	210
TOTAL	404	412

With the extra instructors, six additional flight events are scheduled for the week. These schedule results align with VT-22's statement that instructors are a limiting resource for scheduling.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

TEST-2 serves as an improvement from TEST that provides VT-22 with useable schedules in a quick timeframe relative to manual scheduling. The considerations of weather, combined flight events, and student currency allow VT-22 to use TEST-2 outputs as their daily schedules with minimal tweaking. For the week's data VT-22 provided for comparison, TEST-2 schedules more overall events than the manual schedules created at VT-22 while utilizing combined flight events. Each combined event saves the instructor and student about one hour of briefing time, so the use of 20 combined events in one week is a significant savings.

A. POTENTIAL FOLLOW-ON WORK

1. Second-Pass Optimization Model

Future research may improve the schedules generated by TEST-2 using a "second-pass" optimization model with a finer time resolution. Recall that when constructing the input data for our discrete-time model, we rounded each event's time requirement up rather than down in order to achieve a conservative solution. This conservative solution could be made more time-efficient via a second-pass optimization model. Such a model would use the output of TEST-2 as a "warm start," then allow for small shifts in the schedule. This model's primary goal would be to improve the schedule generated by TEST-2 by "packing" the schedule to reduce idle time for instructors and students. This reduction in idle time could be accomplished by penalizing long periods of time between events for instructors and students.

2. Integration with TSHARP

TEST-2 is not integrated with VT-22's scheduling system, TSHARP. If TEST-2 were to be integrated with TSHARP, the schedulers would save time because they would not have to manually input as much data for each run of TEST-2. There are currently 73 students and 38 instructors at VT-22 and manually entering non-availabilities for each of them is a time-consuming process.

3. Distinguishing Lead Instructors

Neither TEST nor TEST-2 distinguishes lead instructors in events that require a lead instructor. In multiplane events, only certain instructors are qualified to serve as the lead. Incorporating this distinction into the model would improve usability of the schedules created.

4. Improvement of Interface

Although we created an interface for VT-22 to run TEST-2, it has room for much improvement. One main improvement to the interface would be to provide a way to enter the manual inputs on the interface rather than directly into the Excel spreadsheet. Another possible improvement to the interface would be to provide an option for students and instructors to simply enter their name and see what they are scheduled for without having to read through the entire schedule.

5. Faster Solve Time

The weekly model of TEST-2 currently requires about four hours to solve. While this is still an improvement over creating a schedule over the course of a whole workday, options for reducing the model solve time should be explored. A reduction in solve time would make the weekly version of TEST-2 more useable.

B. RECOMMENDATIONS

TEST-2 is a useable tool that will help VT-22's schedulers create schedules in a more efficient manner which will help reduce student overall time-to-train. We recommend VT-22 primarily use the daily version of TEST-2, because it has a much quicker run time than the weekly model. However, VT-22 should run the weekly model at the end of each week in order to get a general outlook for the week ahead. The weekly model could also be run more frequently in a "rolling horizon" fashion during the week, with the initial portion of the planning horizon fixed, and later portions open to optimization.

LIST OF REFERENCES

Eckstrand E (2020) TEST Flight Schedule Planner. Accessed November 19, 2020.

Frazier M, Harris K, Loftus J, Pham B (2019) Training Air Wing Two: Time to train. Unpublished capstone project.

Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) *Pyomo—Optimization Modeling in Python*. Second Edition. Vol. 67. (Springer: Berlin).

Hart WE, Watson JP, Woodruff DL (2011) "Pyomo: Modeling and solving mathematical programs in Python." *Mathematical Programming Computation* 3(3), 219–260.

IBM (2018) *IBM ILOG CPLEX 12.10 User's Manual* (IBM ILOG CPLEX Division, Incline Village, NV).

Jacobs R (2014) Optimization of daily flight training schedules. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/41396

Meditz M (2019) Optimizing training event schedules at Naval Air Station Kingsville. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/64027

Naval Air Training Command (2019, Nov 18) T-45 Combined multi-service pilot training system. CNATRAINST 1542.167B. Department of the Navy. https://www.cnatra.navy.mil/local/docs/instructions/1542.167B.pdf.

Simpson, S (2020) VT-22 scheduling data via personal communication, November 12.

Slye R (2018) Optimizing training event schedules at Naval Air Station Fallon. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/58370

U.S. Department of Transportation, Federal Aviation Administration (2019) *Aeronautical Information Manual: Official Guide to Basic Flight Information and ATC Procedures,* http://www.faraim.org/aim/aim-4-03-14-470.html

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California