

---

*sOptimization of the Class Scheduling  
Process at the Federal Aviation  
Administration Academy Headquarters  
in Oklahoma City, Oklahoma*

---

~~An~~ Oklahoma State University  
School of Industrial Engineering and Management  
Senior Design Project Report

James Darling

Daniel Woods

Brandon Lee

Faculty Mentor Dr. Farzad Yousefian

Spring 2017



Submitted in partial fulfillment of the requirements of the course, IEM 4913, Senior Design and the Baccalaureate degree requirements of the School of Industrial Engineering and Management at Oklahoma State University

## Executive Summary

Formatted: Heading 1, Space After: 12 pt

### Executive Summary

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt

The Federal Aviation Administration (FAA), headquartered in Washington DC, is the federal agency responsible for the operation of civil aviation in the United States. Air Traffic Controllers are the frontline FAA employees upholding this mission by directing the flow of airliner traffic, monitoring flight patterns, and resolving conflicting flight paths. The FAA Academy, located at Mike Monroney Aeronautical Center in Oklahoma City, is currently the FAA's primary location for training Air Traffic Controllers. Newly hired controllers must complete two initial courses: a class in Air Traffic Basics and one of three advanced training classes to acquire a specialty. Each year the FAA must produce an adequate number of Air Traffic Controllers to ensure their primary goal of maintaining a safe National Air Space (NAS). To ensure they produce enough number of students the FAA wishes to increase student throughput, i.e., the number of students that pass both Air Traffic Basics and their specialized course section.

The senior design team for this project consist of three members: James Darling, Daniel Woods, and Brandon Lee. The senior design team used the following four phases to develop a working solution to the FAA's scheduling problem: Planning Phase, Math Model Phase, MATLAB Model Phase, and GUI Phase. Each phase consist of a number of goals with varying priorities and while most of the goals were met some risk mitigation was implemented to ensure maximum effectiveness of the end product. This risk mitigation resulted in the use of two different software packages, Gurobi and MATLAB.

In order to increase student throughput and thus solve the FAA's problem the senior design team looked at the scheduling process. The current process to schedule course sections was completely manual and relied on previous manager experience. Each manager had a list of things they needed to consider when creating the schedule. The result of these things was a schedule that did not generate largest number of students possible.

Formatted: Font: 12 pt

The senior design team produced three potential solutions to the FAA's problem. The first solution was an automated schedule. This solution would mimic current practices and result in man hours saved. The second solution was process improvement. This would involve standardizing the current process managers used to create the schedule and would result in some man hours saved and an easier to read schedule with standardized features. The final Solution was optimization in the form of a Mixed Integer Program (MIP). This would result in

man hours saved and a schedule that would provide more classes and thus students. The final solution was chosen because it addressed the issue of student throughput.

After creating two programs, one with MATLAB and one with Gurobi, the senior design team decided some comparisons between the two would be necessary to determine the best fit for the FAA. The primary difference between the two software's is that Gurobi was built with optimization in mind and MATLAB was built with manipulation of matrices in mind. This difference resulted in Gurobi performing faster and with less intense needs (MATLAB needed a super computer).

The senior design team recommends the Gurobi/python software package. It is understood that the FAA needs to first approve such software before it can purchase a license and begin use. However Gurobi and software similar to it have the ability to (1) increase maximum student throughput by 109 students, (2) reduce man hours spent from 50 hours to four hours , and (3) provide solutions to similar problems such as this.



## Table of Contents

Executive Summary .....	2
1. Background .....	885
2. Methodology .....	12427
2.1 Planning Phase .....	13138
2.2 Math Model Phase .....	14148
2.3 MATLAB Model Phase .....	15159
2.4 GUI Phase .....	15159
2.5 Risk Management .....	15159
3. Current State .....	164611
3.1 Scheduling .....	232311
3.2 Issues in the Scheduling Process .....	242413
4. Solution Alternatives .....	272715
4.1 Automation .....	272715
4.2 Process Improvement .....	282816
4.3 Optimization .....	282816
4.4 Solution Details .....	292916
5. Optimization Model .....	313117
5.1 Sets .....	323218
5.2 Parameters .....	333319
5.3 Decision Variable .....	333319
5.4 Objective Function .....	343419
5.5 Constraints .....	343420
6. Technical Alternatives .....	434322
6.1 MATLAB Model Composition .....	434322

Formatted: Space After: 12 pt

6.2	MATLAB Model Performance .....	444423
6.3	Gurobi Model Composition .....	454524
6.4	Gurobi Model Performance .....	464625
7.	Benefits .....	474726
7.1	Time Commitment .....	474726
7.2	Student Throughput .....	474726
Appendix A: MATLAB Model .....		484827
Appendix B: Gurobi Model .....		565533
Appendix C: Input Tables (.csv) .....		666541
Acknowledgements .....		686844
1.	Background .....	3
1.1	FAA Academy .....	3
2.	Methodology .....	4
2.1	Model .....	5
2.2	MATLAB Implementation .....	8
Figure 1: An Example API Call .....		8
2.3	Graphical User Interface .....	9
3.	Current State Analysis .....	9
3.1	Scheduling .....	9
3.2	Issues in the Scheduling Process .....	10
3.2.1	Resources Consumed .....	10
3.2.2	Cost of Class Size .....	10
3.2.3	Unscheduled Class Cancelations .....	11
3.2.4	No Standardized Schedule Displays .....	11
4.	Solution Alternatives .....	12

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Space Before: 12 pt, After: 12 pt

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Font: Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted**

**Formatted**

**Formatted**

**Formatted**

4.1 Automation..... 12

4.2 Process Improvement..... 13

4.3 Optimization..... 13

5. Recommendations and Benefits..... 14

5.1 Recommendations..... 14

5.2 Benefits..... 14

Appendix..... 14

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

**Formatted:** Default Paragraph Font, Check spelling and grammar

## 1. Background

### 1. Background

The Federal Aviation Administration (FAA), headquartered in Washington DC, is the federal agency responsible for the operation of civil aviation in the United States. Founded by congress in 1958 after several mid-air collisions of passenger airliners, the agency ensures safe passage through the National Airspace System (NAS). Air Traffic Controllers are the frontline FAA employees upholding this mission by directing the flow of airliner traffic, monitoring flight patterns, and resolving conflicting flight paths.

#### 1.1 FAA Academy

The FAA Academy, located at Mike Monroney Aeronautical Center in Oklahoma City, is currently the FAA's primary location for training Air Traffic Controllers. Newly hired controllers must complete two initial courses: a class in Air Traffic Basics and one of three advanced training classes to acquire a specialty. A description of the three, advanced training course sections can be seen in Table 1.

*Table 111: Air Traffic Controller Advanced Course Section(s)*

Course Section	Description
----------------	-------------

Formatted: Space After: 12 pt

Formatted: Heading 1, Space After: 12 pt, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

Formatted: Font: (Default) +Body (Calibri), 12 pt

Formatted: Font: 12 pt

Formatted: Centered, Space Before: 12 pt, After: 12

Formatted: Font: 11 pt

Formatted: Caption, Centered, Space Before: 12 pt, After: 12 pt, Keep with next

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li



<b>Tower</b>	Responsible for aircraft that taxi, take off, and land at an airport.
<b>Radar Tower Facility (RTF)</b>	Responsible for aircraft that enter air space surrounding an airport as they prepare to ascend or descend
<b>En Route</b>	Responsible for aircraft in between airports that have established cruising altitude

- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Table 1: Air Traffic Controller Advanced Course Section(s)

Each course section has access to their own set of class rooms and simulation labs, and is staffed by instructors, remote pilot operators (RPO), and evaluators. Instructors are provided by a third-party service contracted by the FAA, while RPO's and evaluators are employees of the FAA.

The FAA Academy is responsible for graduating a specific quota of specialized Air Traffic Controllers. The specialty of each of these Air Traffic Controllers is determined by which of the advanced training classes they took. The national headquarters of the FAA sets these quotas based on projected controller needs within the next 10 years. Table 2 shows the quota for each of the 2017 advanced courses ~~for 2017 below~~.

- Formatted: Space Before: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Centered, Space Before: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 11 pt
- Formatted: Caption, Centered, Space Before: 12 pt, After: 12 pt, Keep with next
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted Table
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Right, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Table 2: Advanced Course Quotas (2017)

Advanced Course	Student Quota
<b>Tower</b>	712

<b>Radar Tower Facility (RTF)</b>	444
<b>En Route</b>	948

Table 2: Advanced Course Quotas (2017)

Currently, the greatest demand by specialty is for En Route controllers. Due to a budget sequestration over the last few years, the Academy has fallen behind quota by over 400 controllers over the last four years. This means that to date there are 400 less Air Traffic Controllers working in America than the FAA projected there should be. Demand for controllers is also projected to increase through 2019. Because of this projected increase and the FAA's inability to meet previous year's quotas, the FAA's primary goal of ensuring the safety of American air space is at risk. To avoid this risk, the FAA wishes to increase its student throughput. Student throughput is the number of students that pass both Air Traffic Basics and their specialized course section.

## 2. Methodology

To approach this scheduling problem, the team utilized mixed integer programming (MIP) to model and optimize the scheduling process. The MIP model was then coded into the MATLABan optimization engine using the Python a programming language, and then paired to a graphical user interface.

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Right, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Right, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Space After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

Formatted: Heading 1, Space After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Tab stops: 4.5", Left + 4.63", Left

## 2.1 MATLAB Implementation

Once the model was completed mathematically, it was programmed into the MATLAB optimization engine using Python. Python interacts with MATLAB the optimization engine using an Application Program Interface (API) call. An example of an API call is shown below. Our application is Client A. The application requests a token from the MATLAB optimization engine, which then returns a token. The application then submits its token with the data back to the API for processing and optimization (referred to in this diagram as “listening on a channel”).

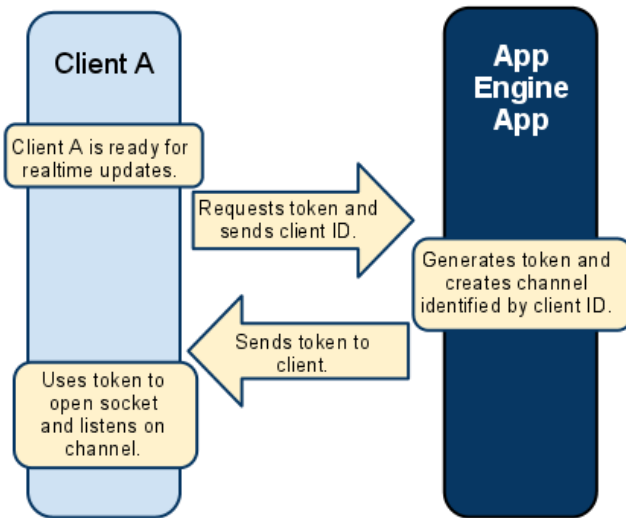


Figure 1: An Example API Call

The function of the Python code is as follows

Formatted: Font: 16 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0.13" + Indent at: 0.38"

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Font: 16 pt

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

- Verify input data
- Organize data for API call
- Interpret MATLAB the engine results
- Organize results for GUI

## 2.2 Graphical User Interface

Once the model was created in Python and validated, it was paired with a GUI, also created using Python. The GUI is paired with the model e Python MATLAB implementation and is involved in the first and last steps of the program. This can be seen in the next figure.

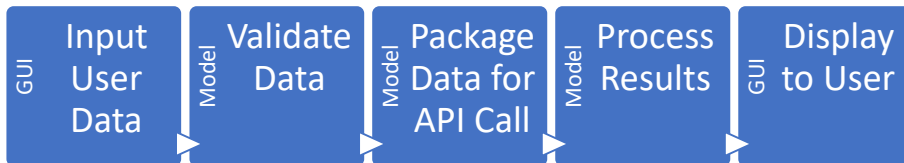


Figure 2: Application Flow Chart

## 2. Methodology

The senior design team for this project consist of three members: James Darling, Daniel Woods, and Brandon Lee. The senior design team used the following four phases to develop a working solution to the FAA’s student throughput problem: Planning Phase, Math Model Phase, MATLAB Model Phase, and GUI Phase. The last section of this methodology discusses the steps taken in each of these phases to minimize risk. Figure 1 below is a Gantt chart of the senior design teams’ methodology with the four previously mentioned phases.

Formatted: Font: 16 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0.13" + Indent at: 0.38"

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

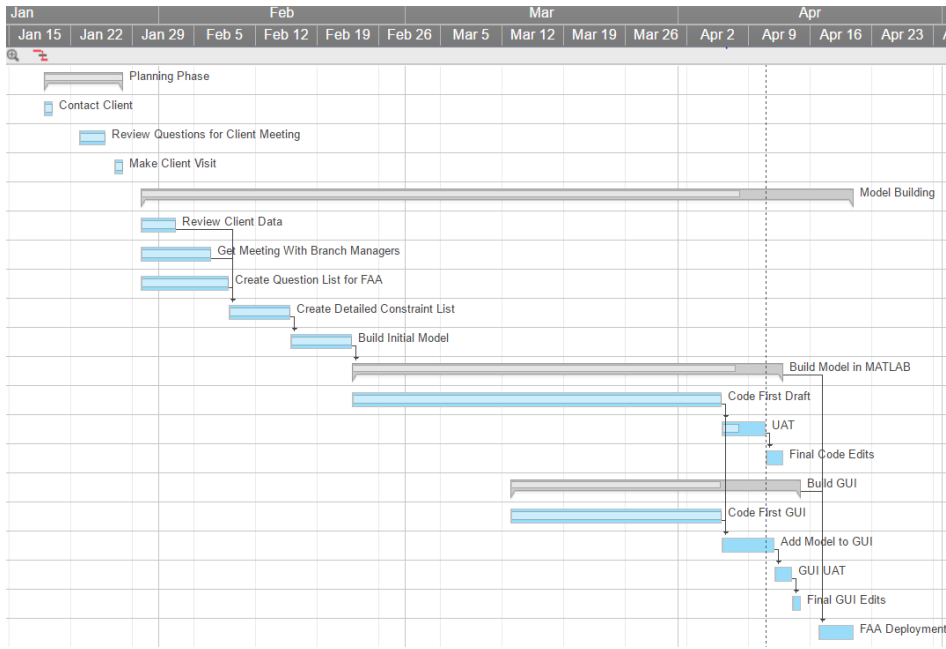
Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 2 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

Formatted: Font: (Default) +Headings (Calibri Light), 16 pt

Formatted: Space After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 2 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"



Formatted: Space Before: 12 pt, After: 12 pt

Figure 1

## 2.1 Planning Phase

The senior design team conducted this phase with limited knowledge of the FAA or their problem. The senior design teams' primary goals during this phase were to make contact with the client, identify their problem(s) and identify possible solutions. A secondary goal was to gather the data necessary to understand the FAA's current situation.

To address the first two goals, the senior design team established contact with the client via email and arranged for a meeting with Dr. Nina Barker, Mr. Jim Duskow and Mr. Wayne Coley at the Mike Monroney Aeronautical center. During this meeting, the senior design team discussed the current situation at the FAA and received a tour of the facility and classrooms. The senior design team's first meeting with the FAA made three things clear. First, the FAA was

dealing trouble graduating enough students. Second, the FAA enjoyed the solution presented by last year's senior design group and was expecting a similar solution from this year's group. Third, another visit would be necessary to meet with staff to discuss the current processes.

To address the last goal of this phase, the senior design team generated and discussed three potential solutions. The senior design team then decided that optimization of the scheduling process would be the preferred solution. Upon deciding this, the senior design team discussed amongst themselves and Ms. Nina Barker potential software packages for this solution. The FAA informed the senior design team at this point that any software package would work and so they made the following list of potential software packages to be considered during the next phase:

- Express
- Gurobi
- MATLAB

## 2.2 Math Model Phase

The senior design team's two primary goals during this phase were to create a working Mixed Integer Programming (MIP) model and come to a decision on which software package to use. Before beginning work on the model, the senior design team needed to fully understand the process being used at the FAA to create schedules. To do this, the senior design team made another visit to the Mike Monroney Aeronautical center. This time they met with several section managers who were in charge of creating the various course section schedules. This meeting provided the senior design team with enough knowledge on the current process to begin work on the model. To facilitate the creation of the model, the senior design team met with faculty adviser Dr. Yousefian. With his help, the senior design team devised a series of steps to create a proper working model. These steps included: Create a list of necessary constraints, create variables, define parameters, define an objective function and verify the model.

To address the second objective, the senior design team began looking into licensing options for software packages to see which software packages would be the least expensive for the FAA to implement. Upon further investigation, the senior design team discovered that the FAA already had a license of MATLAB. This eliminated the need to consider other software packages due to the potential savings of not purchasing a new license.

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

### 2.3 MATLAB Model Phase

The senior design team's goals in this phase were to create a working model in MATLAB and to ensure that this model was usable by the end users. The senior design team decided that the end users in this case were course section managers. To ensure the first goal, the senior design team set up regularly meetings with Dr. Yousefian to help facilitate the creation of the model. His help ensured that the senior design team created a model using logical steps and the correct functions. The feedback from other professors, namely Dr. Buchanan and Dr. Baski, also ensured that the senior design team was taking the correct approach to the MATLAB model. The senior design team did face difficulty ensuring the second goal of this phase, as the final version of the MATLAB model ran into many errors when running. This problem is discussed in further detail in the Risk Management section of this report.

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

### 2.4 GUI Phase

The last phase had two objectives. The first was to create a working graphical user interface (GUI). The second objective was to make that GUI as accessible as possible. To clarify, accessibility in this context means that the user requires as little training/instruction as possible to be able to use the software. To ensure both objectives the senior design team devised the following steps:

- Code the first GUI
- Add the MATLAB model to the GUI
- User Acceptance Test (UAT)
- Final GUI Edits

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

### 2.5 Risk Management

This section will discuss the steps the senior design team took to mitigate risk during the execution of this project. One of the primary concerns the senior design team had was the functionality of the final product. As programming began in MATLAB this concern grew as the senior design team realized there was a possibility that this software package might not be very useful to the end users. There were two reasons for this. First, a program written in MATLAB would take significantly longer to run. The senior design team came to this conclusion after discussing MATLABs merit as an optimization engine with Dr. Buchanan. The second reason is that MATLAB was likely to require a super computer to run. The senior design team made this conclusion after discussion with Dr. Yousefian. The senior design team felt these two reasons posed a significant enough threat to the programs functionality and so they developed measures to mitigate this risk.

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

The senior design team decided that the best mitigation strategy for this risk was to create a backup program using the Python/Gurobi software package. The senior design team made this decision for two reasons. First, the senior design team had more collective experience with the Python/Gurobi software package. The second reason is that the Python/Gurobi software package would require less time to create a working program. This meant that the senior design team could confidently generate a backup model without compromising the MATLAB program.

Formatted: Font: 12 pt

### 3. Current State~~Current State Analysis~~

Formatted: Space After: 12 pt, Line spacing: Multiple 1.15 li

In its current state, the FAA fails to produce enough students to satisfy their goal of providing safe civilian air space. To alieviate this problem the FAA wishes to increase student throughput or the number of students that pass both Air Traffic Basics and their specialized course section. The FAA captures this data by keeping track of the number of students participating in each course offering and in turn the number of students in each course section. Student throughput



is currently measured as the number of students each course section is pushing through each fiscal year. Since student throughput cannot exceed the max capacity of any course section, this capacity represents a constraint on student throughput. Max capacity is the maximum amount of students a course section can produce. This number is calculated by section managers. Currently there are no course sections with a throughput equal to or near max capacity.

There are two ways to increase student throughput. The first is to improve current scheduling practices to reach max capacity. The second is to increase max capacity. The second method requires changes that require a larger capital investment. Such changes include but are not limited to the purchasing of more classrooms, an increase in the number of FAA evaluators available, or a change in the number of students that can be taught per course offering.

## — Scheduling

### 2.1

The process to schedule course sections is completely manual and relies on previous manager experience. The manager for each section spends between 8-12 hours developing their section schedule in Microsoft Excel for the next fiscal year. This process take place in June, four months before the start of the next fiscal year. Every year the FAA provides the federal government with information regarding the maximum student throughput each course section can achieve. The federal government uses this number to help determine the quota for each course section in a given fiscal year

The schedule for each course section is currently created by the managers responsible for that specific course. The managers build their schedule to maximize student throughput. Student throughput is defined as the number of students that pass both Air Traffic Basics and their specialized course section. In addition to maximizing student throughput these managers have a list of things they need to keep in mind when building their schedules:

- Each course section is offered many times throughout the year and each iteration is called an offering.
- Each offering can be conducted during the day shift (7am-3:30pm) or the night shift (3:30pm-Midnight)
- Each offering consists of different classes which can be taught in different sets of rooms. For example, an offering of basics has five different classes: Academics, Lab 1, Personal Assessment, Lab 2, Final Assessment.
- Each class must be taught in an appropriate room.

Formatted: Font: 12 pt

Formatted: Font: (Default) +Body (Calibri), 12 pt

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, Line spacing: Multiple 1.15 li

Formatted: List Paragraph, Space Before: 12 pt, Line spacing: Multiple 1.15 li, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

- En route offerings are 59 days long and have six different classes
- Tower offerings are 36 days long and have six different classes
- RTF has two offering types. The first is the normal offering and the second is an advanced portion that takes place directly after the first. Students are not required to take the second offering type and it is scheduled according to student demand.
- RTF's first offering is 21 days long and has three different classes
- RTF's second offering is 15 days long and has three different classes
- 
- Basics offerings have two different class types: Basics tower and basics en route.
- Basics offerings are both 25 days long and have four different classes
- Every offering of basics is paired with an offering of its associated type (so so basics tower pairs with a tower offering and basics en route with an en route offering). The same group of students partake in both classes each pairing. These classes must be scheduled back to back with no gap in between.
- Each offering consists of different classes which can be taught in different sets of rooms. For example, an offering of basics has five different classes: Academics, Lab 1, Personal Assessment, Lab 2, Final Assessment.
- Each class must be taught in an appropriate room.
- The last class for every course section is a Final Assessment. This assessment requires the supervision of evaluators (FAA employees) assessment. En route and basics have their own pool of 7 evaluators and RTF and Tower share a pool of 7.
- Because one Final Assessment requires the attention of 6 evaluators no more than one Final Assessment can take place on any given day for offerings sharing a pool of evaluators.
- Each offering needs to be placed with consideration to variability. Things such as snow days, fires, etc. can cause students to miss a day of class. Because of this, managers try to leave around one day in between each offering.

Each section offers the same training many times throughout the year. The training can be conducted during the day shift (7am-3:30pm) or the night shift (3:30pm-Midnight).

The process to schedule course sections was completely manual and relied on previous manager experience. The manager for each section would spend between 8-12 hours developing their section schedule for the next fiscal year. This would take place in June, four months before the start of the next fiscal year.

The quota for each course section determines to a certain extent how constrained each of those schedules will be. Meeting a higher quota means scheduling more classes and more classes in the schedule means less room for variability such as snow days or the need to add

Formatted: List Paragraph, Space Before: 12 pt, Line spacing: Multiple 1.15 li

Formatted: List Paragraph, Space Before: 12 pt, Line spacing: Multiple 1.15 li, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Space Before: 12 pt, Line spacing: Multiple 1.15 li

another class. Because schedule flexibility diminishes with an increase in students, the schedules were developed in an order dependent on the section with greatest current demand. For 2017 fiscal year, En Route carried the greatest demand with a quota of 968 students. Air Traffic Basics were then scheduled to terminate at the beginning of the En Route class for students in that track. Tower and RTF were then completed to align their start dates with the end of Air Traffic Basics. The goal of this style of back scheduling is to minimize the amount of time students sit idle between classes (students are salaried FAA employees, and all idle time still incurs costs).

After all the section schedules are were created, two upper level managers would review the schedules to make sure that the required quotas ar were attained and that the schedules interact correctly as a whole. Scheduling ta ookes approximately two weeks and involves d 12 managers.

[Insert Graphic of time line here]

Between the planning period and start of the fiscal year, managers would have to make more manual edits to the schedule depending on changes in quotas from the FAA and availability of trainees. These E edits usually require an additional eight hours of review by section managers to ensure that certain restrictions are not violated (these restrictions will be covered in the Modeling section).

### 3.2 Issues in the Scheduling Process

#### 3.2.1 Resources Consumed

We identified several issues within the Academy's scheduling process. The first is the amount of man hours consumed to create schedules manually. In total, it takes oook FAA managers nearly 50 man hours over the course of two weeks to manually create and validate the course schedules for each course section in Microsoft Excel. These managers are also expected to continue their usual duties of managing class content and delivery during this time. All time spent away from course management can incur additional labor hours for problem solving by third party instructors and other staff. result in a loss in the overall quality of the courses, as they are not receiving as much attention from management as they could be. This is a problem

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: (Default) +Body (Calibri), 12 pt

Formatted: Font: 12 pt

Formatted: Font: (Default) +Body (Calibri), 12 pt

Formatted: Font: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

because any compromise to course quality will affect the quality of the students produced and this can have a negative effect on the safety of American air space.

3.2.2 Cost of Class Size

The ideal instructor to student ratio is 2 to 3. Forming classes that are off ratio can incur additional instructor costs will increase the cost per student. The reason for this that instructors and students do not have a 1:1 ratio (as one instructor is often responsible for multiple students) and so while the number of students may drop, the number of instructors might not. Another factor that contributes to this is the cost of the facility being used. The more students per facility the lower the per student cost. While the exact baseline and value of the cost for an 18 student class are confidential, the percentage increase of costs for deviating from this predetermined optimum are listed in the table below. Please note that 18 is the optimal number because it is the maximum number of students most classes can accommodate without breaking fire safety standards.

Class Size	Cost Increase
18	-
17	9.5%
16	11.2%
15	10.0%
14	16.0%
13	18.0%

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted Table
...

Formatted Table
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

Formatted
...

20

12	13.5%
11	21.0%
10	29.0%
9	20.0%
8	30.0%
7	40.0%
6	42.0%

Table 3: Cost of Deviating from Optimum Student Ratio

Because each of the schedules were massive, manually reviewed spreadsheets, it was very difficult for managers to tell the best way to create classes of 18. Utilizing years of previous experience, the managers could create most classes with a size of 18, however there were still numerous class sizes of 12. This increases costs some real data here, need to calculate it.

### 3.2.3 Unscheduled Class Cancellations

The third issue involves the cost of unpredictable class cancellations. The Academy plays a delicate balancing game between packing course sections full of classes and adding buffers to account for unscheduled class cancellations. Events like snow days, tornado warnings, bomb threats, and fire drills can cause classes to miss anywhere from a few

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

hours to a couple days. If a schedule is built without any buffers, these events can cause timely setbacks and cause a massive backlog in the section schedule. However, each buffer costs the Academy student throughput.

### 3.2.4 No Standardized Schedule Displays

The final issue is the confusion created by a lack of standardized schedule formatting. Each schedule created by the section managers was color-coded in a different manner. When the completed schedules were submitted to the upper level managers, they would have to spend additional labor hours identifying which colors and boxes meant different things.

**Formatted:** Font: (Default) +Headings (Calibri Light), 13 pt

**Formatted:** Centered, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

**Formatted:** Font: (Default) +Headings (Calibri Light), 13 pt

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Somewhere in here you should summarize the problem statement; either before or after you list the issues. I don't suggest adding a separate section called problem statement, but summarize for the reader what you're solving for, and state it clearly.

**Formatted:** Font: (Default) +Headings (Calibri Light), 13 pt

Figure 3: Spreadsheet example

**Formatted:** Centered, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

### 3. Solution Industrial Engineering Approach Alternatives

#### 3.1 Scheduling

The process to schedule course sections is completely manual and relied on previous manager experience. The manager for each section spent between 8-12 hours developing their section schedule in Microsoft Excel for the next fiscal year. This process takes place in June, four months before the start of the next fiscal year. Every year the FAA provides the federal government with information regarding the maximum student throughput each course section can achieve. The federal government uses this number to help determine the quota for each course section in a given fiscal year.

The schedule for each course section is currently created by the managers responsible for that specific course. The managers build their schedule to maximize student throughput. In addition to maximizing student throughput these managers have a list of things they need to keep in mind when building their schedules:

- Each course section is offered many times throughout the year and each iteration is called an offering.
- Each offering can be conducted during the day shift (7am-3:30pm) or the night shift (3:30pm-Midnight)
- Each offering consists of different classes which can be taught in different sets of rooms. For example, an offering of basics has five different classes: Academics, Lab 1, Personal Assessment, Lab 2, Final Assessment.
- Each class must be taught in an appropriate room. An appropriate room is any room with the necessary equipment to teach a class.
- En Route offerings are 59 days long and have six different classes
- Tower offerings are 36 days long and have six different classes
- RTF has two offering types. The first is the normal offering and the second is an advanced portion that takes place directly after the first. Students are not required to take the second offering type and it is scheduled according to student demand.

**Formatted:** Font: 13 pt

**Formatted:** Space After: 12 pt, Line spacing: Multiple 1.15 li

**Formatted:** Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

- RTF's first offering is 21 days long and has three different classes
- RTF's second offering is 15 days long and has three different classes
- Basics offerings have two different class types: Basics tower and basics en route.
- Basics offerings are both 25 days long and have four different classes  
Every offering of basics is paired with an offering of its associated type (so basics tower pairs with a tower offering and basics en route with an en route offering). The same group of students partake in each pairing. These classes must be scheduled back to back with no gap in between.
- The last class for every course section is a Final Assessment. Each assessment requires the supervision of six evaluators (FAA employees), who act as proctors for the assesment. En route and Basics have their own pool of 7 evaluators and RTF and Tower share a pool of 7.
- Because one Final Assessment requires the attention of 6 evaluators no more than one Final Assessment can take place on any given day for offerings sharing a pool of evaluators.
- Each offering needs to be placed with consideration to variability. Things such as snow days, fires, etc. can cause students to miss a day of class. Because of this managers try to leave around 1 day in between each offering.

After all the section schedules are created, two upper level managers review the schedules to make sure that the required quotas are attained and that the schedules interact correctly as a whole. Scheduling takes approximately two weeks and involves 12 managers.

Between the planning period and start of the fiscal year, managers have to make more manual edits to the schedule depending on changes in quotas from the FAA and availability of trainees. These edits usually require an additional eight hours of review by section managers to ensure that certain restrictions are not violated (these restrictions will be covered in the Modeling section). This is necessary because with the number of restrictions that managers need to take into consideration, it is extremely easy to make an oversight.

### 3.2 Issues in the Scheduling Process

We identified several issues within the Academy's scheduling process. The first is the amount of man hours consumed to create schedules manually. In total, it takes FAA managers nearly 50 man-hours over the course of two weeks to manually create and validate the schedules for each course section in Microsoft Excel. These managers are also expected to continue their usual duties of managing class content and delivery during this time. All time spent away from course management can result in a loss in the overall quality of the courses, as they are not receiving as much attention from management as they could be. This is a problem because any

**Formatted:** Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"



compromise to course quality will affect the quality of the students produced and this can have a negative effect on the safety of American air space.

The ideal instructor-to-student ratio is 2 to 3. Forming classes that are off-ratio will increase the cost per student. The reason for this that instructors and students do not have a 1:1 ratio (as one instructor is often responsible for multiple students) and so while the number of students may drop, the number of instructors might not. Another factor that contributes to this is the cost of the facility being used. The more students per facility the lower the per student cost. While the exact baseline and value of the cost for an 18-student class are confidential, the percentage increase of costs for deviating from this predetermined optimum are listed in the table below. Please note that 18 is the optimal number because it is the maximum number of students most classes can accommodate without breaking fire safety standards. The exception to this are lab rooms with a maximum capacity of 12 students. For these rooms the optimal number is 12. This number when combined with the utilization rate of the rooms determines the student throughput of any given fiscal year.

*Table 333: Cost of Deviating from Optimum Student Ratio*

<u>Class Size</u>	<u>Cost Increase</u>
<u>18</u>	<u>-</u>
<u>17</u>	<u>9.5%</u>
<u>16</u>	<u>11.2%</u>
<u>15</u>	<u>10.0%</u>
<u>14</u>	<u>16.0%</u>
<u>13</u>	<u>18.0%</u>
<u>12</u>	<u>13.5%</u>

- Formatted: Font: 11 pt
- Formatted: Caption, Centered, Keep with next
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted Table
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li

<u>11</u>	<u>21.0%</u>
<u>10</u>	<u>29.0%</u>
<u>9</u>	<u>20.0%</u>
<u>8</u>	<u>30.0%</u>
<u>7</u>	<u>40.0%</u>
<u>6</u>	<u>42.0%</u>

Because the pass/fail rate of the classes and in the instance of bargaining units the demand of some classes, not all classes are composed of the maximum number of students. This can be a problem as class sizes, due to either of the aforementioned things, approach smaller numbers. As table 3 shows the smaller the class size the higher the overhead cost per student is. Because of this cost increase, some classes get canceled. Please note that bargaining units will be explained further in section 5 of this report.

The third issue involves the cost of unpredictable class cancelations. The Academy plays a delicate balancing game between packing course sections full of classes and adding buffers to account for unscheduled class cancelations. Events like snow days, tornado warnings, bomb threats, and fire drills can cause classes to miss anywhere from a few hours to a couple days. If a schedule is built without any buffers, these events can cause timely setbacks and cause a massive backlog in the section schedule. However, each buffer costs the Academy student throughput.

The final issue is the confusion created by a lack of standardized schedule formatting. Each schedule created by the section managers was color-coded in a different manner. When the completed schedules were submitted to the upper level managers, they would have to spend additional labor hours identifying which colors and boxes meant different things. Figure 2 demonstrates the color coding used in the schedules.

- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt
- Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li
- Formatted: Font: 12 pt

Formatted: Font: 12 pt

Figure 2: Spreadsheet example

Formatted: Font: 12 pt  
Formatted: Keep with next

#### 4. Solution Alternatives

The senior design team considered two couple of alternative approaches to the scheduling problem. The first consideration was automation and the second was process improvement. While both had the potential to solve the FAA's problem the solution with the highest potential was an optimization approach.

##### Automation

##### 4.1 Automation

Automation of the FAA's scheduling process would involve the creation of a program similar to the one created in our solution. This program would then emulate the process that managers currently undergo to create a schedule. That is to say that it would take one class offering, find the first available opening in the schedule and it would place it there. It would then continue this process until it ran out of available spaces in a given fiscal year.

Such a program would have many benefits. First of all, it would save the managers' time. No longer would they need to spend hours creating an initial draft of a schedule. Second, it would allow the schedule to be created with strict adherence to a set of rules. An example of one such rule would be avoiding scheduling classes during federal holidays. Third, it would allow for

Formatted: Heading 1, Space After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 2 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

Formatted: Font: 12 pt  
Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

managers to easily edit the schedule should unforeseen circumstances close the academy for a day.

This solution does have drawbacks however. First, the set of rules that the program would be allowed to adhere to would be somewhat limited when compared to an optimized schedule. Second, the program would only be able to create schedules at or near max capacity. While this is not inherently a problem given the FAA's current situation, it could be further down the line should the FAA need less students.

### 3.2 Process Improvement

#### 4.2 Process Improvement

Process improvement would entail taking the manual process managers go through in excel to create the schedules and standardizing/improving it. The goal here would be to create a step by step process the managers could follow that would allow them to generate a schedule in as little time as possible. The primary benefit offered from this approach is time saved creating the schedules. It is also possible to make design the process in such a way that it allows for managers to more easily identify openings in the schedule. This means that there could be an impact on the student throughput with the method.

While this alternative has some potential, it has many problems. First and foremost is that the primary benefit of this alternative would be time saved creating the schedule. This does not align with problem of the current state which is student throughput. Though it is true that student throughput could be improved this way, the potential for improvement is small.

### — Optimization

#### 4.3 Optimization

Optimization involves the creation of a math model to solve our scheduling problem. Such a model would seek to maximize or minimize some aspect of the schedule. It would also adhere to a certain set of constraints. The end result would be a schedule that is either optimal or near optimal.

This solution has many potential benefits. First and foremost is the fact that student throughput can be maximized with this method. This would allow us to generate the best solution for the FAA's primary problem (meeting student quotas). The second benefit is that we can build the

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

model to adhere to many different constraints. This allows us to create a schedule that follows a set of rules, similar to the automation solution. The last benefit is the fact that such a model can easily be run through programs such as MATLAB. This means that managers will not have to waste ~~nearly~~ as much time creating/editing the schedule.

#### 3.44.4 Solution Details

~~The final team decision was to decided on using~~ an optimization solution that would run in ~~MATLAB~~ the optimization engine. This solution required the development of a Mixed Integer Programming model with maximization of student throughput as the objective. ~~For this reason, the team created a MIP model with maximizing student throughput as the objective.~~

~~The team documented~~ The following ~~the~~ list of restrictions and limitations the FAA had with scheduling ~~was converted to mathematical constraints for the MIP model, and converted them to mathematical constraints for the MIP model.~~ These restrictions include:

- All Classes
  - A class occupies a room
  - A class takes place during a shift on a given day
  - Classes must follow a certain sequence (class one must precede class two etc.)
  - Each day has two shifts
  - Each course section is composed of offerings
  - Each offering is composed of classes
  - The last class of each offering is a final assessment
  - Each final assessment requires six evaluators
  - A course section cannot use more evaluators on any given day than are available
  - Offerings cannot have more than three consecutive weeks of night shift classes
  - Offerings cannot switch from night shift to day shift in the middle of the week (has to take place over the weekend)
- Air Traffic Basics

Formatted: Font: 12 pt

- Students cannot have more than three consecutive weeks of night shifts. The last day of every basics en route offering must take place the day before the first day of an en route offering
- the last day of every basics tower offering must take place the day before the first day of a tower offering
- 
- RTF and Tower
  - Night classes incur additional costs for instructors. Share a pool of seven evaluators

The mathematical model is listed out in the following sections.

Formatted: Font: 12 pt

Formatted: Font: 12 pt

### Sets

- $A_{r,c,s} = \{0, 1\}$ 
  - This equals 1 if a room  $r$  can host a class  $c$  in section  $s$ . This equals 0 otherwise.
- $C = \{1 : \textit{Academics}, 2 : \textit{Orientation}, 3 : \textit{Lab}, 4 : \textit{Exam}\}$ 
  - Associated with class
  - For those sections that do not have an orientation, Index 2 is treated as a dummy value.
- $D = \{1, \dots, d_Y\}$ 
  - Associated with day
  - Where  $d_Y$  is the number of days in fiscal year  $Y$
- $D_B = \{1, \dots, d_B\}$ 
  - Associated with days that a class cannot begin on for bargaining units
  - Includes start days where the associated class would end on an unallowed day for bargaining units
  - Where  $d_B$  is the number of days on which a class cannot begin
- $O = \{1, \dots, m_O\}$ 
  - Associated with offering
  - Where  $m_O$  is the maximum number of offerings for the largest section
- $O_B = \{1, \dots, o_B\}$ 
  - Associated with offering
  - Where  $o_B$  is the number of bargaining unit offerings
- $R = \{1, \dots, m_R\}$ 
  - Associated with room
  - Where  $m_R$  is the maximum number of rooms for the largest section
- $S = \{1 : \textit{Basics}, 2 : \textit{Tower}, 3 : \textit{RTF}, 4 : \textit{EnRoute}\}$ 
  - Associated with section
- $T = \{0 : \textit{Day}, 1 : \textit{Night}\}$ 
  - Associated with shift/time

- Classes cannot begin on Mondays
- Classes cannot begin the day after federal holidays
- Classes cannot end on Fridays
- Classes cannot end the day before federal holidays
- Between RTF and Tower, only one class can have an evaluation on a given day
- Cannot have a difference in instructors needed greater than 20 between two consecutive days

◆ En Route

- Only 4 classes can use the labs on a given day
- Only one class section can hold an examination on a given day
- Classes cannot begin on Mondays
- Classes cannot begin the day after federal holidays
- Classes cannot end on Fridays
- Classes cannot end the day before federal holidays
- Cannot exceed 130 instructors needed on a given day

Formatted: Font: 12 pt

Formatted: List Paragraph, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, Bulleted + Level: 2 + Aligned at: 0.75" + Indent at: 1"

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

## 5. Optimization Model

The optimization technique known as mixed integer programming (MIP) that our senior design team has selected to address the scheduling problem uses complex mathematical logic to find the optimal solution for a specific objective. The technique and associated language often requires advanced training and knowledge to understand. In order to mediate the knowledge requirement, the following sections thoroughly document and explain the model and its individual components such as data sets, indices, variables, objective function, and constraints. The documentation will provide the FAA scheduling managers and engineers a clear understanding of the model so that the solution can be used in the most beneficial way.

Formatted: Heading 1, Space After: 12 pt, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 2 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

### 5.1 Sets

An index set is a discrete collection of data values. The following are the nine sets are used throughout the model.

- $A_{r,c,s} = \{0, 1\}$

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

- This equals 1 if a room r can host a class c in section s. This equals 0 otherwise.
- $C = \{1 : \textit{Academics}, 2 : \textit{Orientation}, 3 : \textit{Lab}, 4 : \textit{Exam}\}$ 
  - Associated with class
  - For those sections that do not have an orientation, Index 2 is treated as a dummy value.
- $D = \{1, \dots, d_Y\}$ 
  - Associated with day
  - Where  $d_Y$  is the number of days in fiscal year Y
- $D_B = \{1, \dots, d_B\}$ 
  - Associated with days that a class cannot begin on for bargaining units
  - Includes start days where the associated class would end on an restricted day for bargaining units
  - Where  $d_B$  is the number of days on which a class cannot begin
- $O = \{1, \dots, m_O\}$ 
  - Associated with offering
  - Where  $m_O$  is the maximum number of offerings for the largest section
- $O_B = \{1, \dots, o_B\}$ 
  - Associated with offering
  - Where  $o_B$  is the number of bargaining unit offerings
- $R = \{1, \dots, m_R\}$ 
  - Associated with room
  - Where  $m_R$  is the maximum number of rooms for the largest section
- $S = \{1 : \textit{Basics}, 2 : \textit{Tower}, 3 : \textit{RTF}, 4 : \textit{EnRoute}\}$ 
  - Associated with section
- $T = \{0 : \textit{Day}, 1 : \textit{Night}\}$



- o Associated with shift/time

## 5.2 Parameters

A parameter is a known value that provides a circumstantial characteristic with an associate variable. The nine model parameters are listed below.

- $e_{Total}$  is the total number of evaluators
- $e_{Needed}$  is the needed number of evaluators
- $g_{Total}$  is the total number of RSO's
- $g_{Needed}$  is the needed number of RSO
- $I_{Total}$  is the total number of instructors
- $I_{Needed}$  is the needed number of instructors
- $p_{Total}$  is the total number of RPOs
- $p_{Needed}$  is the needed number of RPOs
- $u_{s,c}$  is the duration of class c in section c

## 5.3 Decision Variable

A decision variable is the quantity which is controlled by the decision-maker.

$$\underline{X_{s,c,o,d,t,r}}$$

X is defined as the decision to start an offering o of class c in section s on day d during time t in room r. X is a binary decision variable.

## 5.4 Objective Function

The objective function is the value the model wishes to optimize.

Maximize

$$\underline{\sum_s \sum_c \sum_o \sum_d \sum_t \sum_r X_{s,c,o,d,t,r}}$$

Maximizes the number of times that a course offering o of class c in section s on day d during shift t in room r occurs within a fiscal year.

**Formatted:** Heading 2, Space Before: 12 pt, After: 12 pt, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

**Formatted:** Heading 2, Space Before: 12 pt, After: 12 pt, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

## 5.5 Constraints

Constraints are defined a real world boundaries that exist within the system being optimized. The ten model constraints are listed and explained below.

$$X_{s,c,o,d,t,r} + \sum_{s' \neq s} \sum_{c' \neq c} \sum_{d'=d} \sum_{o' \neq o} X_{s',c',o',d',t,r} \leq 1$$
$$\forall d, r, t, c' \in C, d' \in D, o \in O, s' \in S,$$

No two offerings of a particular course of a particular section may occur on the same shift of the same day in the same room.

$$e_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$
$$\forall d, t, s = 1, 4, o \in O$$

The number of Evaluators for Basics and En Route used on any particular shift on any particular day cannot exceed the total number of evaluators available.

$$e_{Needed} \sum_{s=2,3} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$
$$\forall d, t, o \in O$$

The number of Evaluators for Tower and RTF used on any particular shift on any particular day cannot exceed the total number of evaluators available. This pool of evaluators is shared between these two sections.

$$g_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq g_{Total}$$
$$\forall d, s, t, o \in O$$

The number of RSO's used on any particular shift on any particular day cannot exceed the total number of RSO's available.

$$I_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq I_{Total}$$

$$\forall d, t, s, o \in O$$


---

The number of instructors used on any particular shift on any particular day cannot exceed the total number of instructors available.

$$P_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq P_{Total}$$

$$\forall d, s, t, o \in O$$


---

The number of RPO's used on any particular shift on any particular day cannot exceed the total number of RPO's available.

$$X_{s,c,o,d,t,r} \leq \sum_{t'} \sum_{r'} X_{s,c+1,o,d+u_{c,s,t'},r'}$$

$$\forall d, o, s, t$$


---

The classes in a given section must occur in order (1 to 2 to 3 to 4). Each class c of section s for shift t in room r must begin the day after the previous class ends.

$$X_{3,c,o,d,t,r} = 0$$

$$\forall d \in D_B, \forall o \in O_B, \forall c$$


---

Bargaining unit offerings cannot start on days that would incur additional cost. Bargaining units are students who have already graduated the academy and have work experience in the field. Currently, only RTF has bargaining units, requiring s=3

$$\sum_o \sum_d \sum_t A_{r,c,s} X_{s,c,o,d,t,r} \geq 0$$

$$\forall c, s, r \in R$$


---

Due to unique technology in each room, certain rooms can only host certain classes.

$$X_{1,4,o,d,t,r} = X_{2,1,o,d+u_{s,c,t},r} + X_{4,1,0,d+u_{s,c,t},r}$$

$$\forall d, o, r, t$$


---

Basics classes must be followed by either an En Route or Tower class the next day.

## Parameters

- $e_{Total}$  is the total number of evaluators
  - $e_{Needed}$  is the needed number of evaluators
  - $g_{Total}$  is the total number of RSO's
  - $g_{Needed}$  is the needed number of RSO
  - $I_{Total}$  is the total number of instructors
  - $I_{Needed}$  is the needed number of instructors
  - $p_{Total}$  is the total number of RPOs
  - $p_{Needed}$  is the needed number of RPOs
  - $u_{s,c}$  is the duration of class c in section c
- 

## Decision Variables

- $X_{s,c,o,d,t,r}$

## Constrants

No two offerings of a particular course of a particular section may occur on the same shift of the same day in the same room.

$$X_{s,c,o,d,t,r} + \sum_{s' \neq s} \sum_{c' \neq c} \sum_{d' = d} \sum_{o' \neq o} X_{s',c',o',d',t,r} \leq 1$$

$$\forall d, r, t, c' \in C, d' \in D, o \in O, s' \in S,$$

The number of Evaluators for Basics and En Route used on any particular shift on any particular day cannot exceed the total number of evaluators available.

$$e_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$
$$\forall d, t, s = 1, 4, o \in O$$

The number of Evaluators for Tower and RTF used on any particular shift on any particular day cannot exceed the total number of evaluators available. This pool of evaluators is shared between these two sections.

$$e_{Needed} \sum_{s=2,3} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$
$$\forall d, t, o \in O$$

The number of RSO's used on any particular shift on any particular day cannot exceed the total number of RSO's available.

$$g_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq g_{Total}$$
$$\forall d, s, t, o \in O$$

The number of instructors used on any particular shift on any particular day cannot exceed the total number of instructors available.

$$I_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq I_{Total}$$
$$\forall d, t, s, o \in O$$

X is defined as the decision to start offering o of class c in section s on day d during time t in room r. X is a binary decision variable.

---

### Constrints

No two offerings of a particular course of a particular section may occur on the same shift of the same day in the same room.

$$X_{s,c,o,d,t,r} + \sum_{s' \neq s} \sum_{c' \neq c} \sum_{d'=d} \sum_{o' \neq o} X_{s',c',o',d',t,r} \leq 1$$

$$\forall d, r, t, c' \in C, d' \in D, o \in O, s' \in S,$$

The number of Evaluators for Basics and En Route used on any particular shift on any particular day cannot exceed the total number of evaluators available.

$$e_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$

$$\forall d, t, s = 1, 4, o \in O$$

The number of Evaluators for Tower and RTF used on any particular shift on any particular day cannot exceed the total number of evaluators available. This pool of evaluators is shared between these two sections.

$$e_{Needed} \sum_{s=2,3} \sum_o \sum_r X_{s,c,o,d,t,r} \leq e_{Total}$$

$$\forall d, t, o \in O$$

The number of RSO's used on any particular shift on any particular day cannot exceed the total number of RSO's available.

$$g_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq g_{Total}$$

$$\forall d, s, t, o \in O$$

The number of instructors used on any particular shift on any particular day cannot exceed the total number of instructors available.

$$I_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq I_{Total}$$

$$\forall d, t, s, o \in O$$

The number of RPO's used on any particular shift on any particular day cannot exceed the total number of RPO's available.

$$P_{Needed} \sum_o \sum_r X_{s,c,o,d,t,r} \leq P_{Total}$$

$$\forall d, s, t, o \in O$$

The classes in a given section must occur in order (1 to 2 to 3 to 4). Each class c of section s for shift t in room r must begin the day after the previous class ends.

---

$$X_{s,c,o,d,t,r} \leq \sum_{o'} \sum_{r'} X_{s,c+1,o,d+u_{c,s},t',r'} \\ \forall d, o, s, t$$

Bargaining unit offerings cannot start on days that would incur additional cost. Currently, only RTF has bargaining units, requiring  $s = 3$

$$X_{3,c,o,d,t,r} = 0 \\ \forall d \in D_B, \forall o \in O_B, \forall c$$

Due to unique technology in each room, certain rooms can only host certain classes.

$$\sum_o \sum_d \sum_t A_{r,c,s} X_{s,c,o,d,t,r} \geq 0 \\ \forall c, s, r \in R$$

Basics classes must be followed by either an En Route or Tower class the next day.

$$X_{1,4,o,d,t,r} = X_{2,1,o,d+u_{s,c},t,r} + X_{4,1,0,d+u_{s,c},t,r}$$

The preliminary model for schedule optimization at the FAA is listed below. After the model, we have listed an explanation of each line.

Maximize  $\sum_{e,d,t,r} X_{e,d,t,r}^-$

$$\sum_e X_{e,d,t,r}^- \leq 1, \forall d, t, r$$

$$\sum_e X_{e,d,t,r}^- = U_e, \forall c, t, r$$

$$\sum_e E_e X_{e,d,t,r}^- \leq E_{total}, \forall d, t, r$$

$$\sum_e N_e X_{e,d,t,r}^- \leq N_{total}, \forall d, t, r$$

$$\sum_e P_e X_{e,d,t,r}^- \leq P_{total}, \forall d, t, r$$

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

$$\sum_c R_c X_{c,d,t,r}^- \leq R_{total}$$

$$A_{r,c} \geq X_{c,d,t,r} \forall c, d, t, r$$

$$B_{c,d,t} X_{c,d,t,r} = 0, c = 1, d \in M, \forall t, r$$

$$B_{c,d,t} X_{c,d,t,r} = 0, c = last, d \in F, \forall t, r$$

$$X_{c,d,t,r} = X_{c+1,d+1,t,r} \forall c, d, t, r$$

$X_{c,d,t,r}$  Binary

Definitions

- $X_{c,d,t,r}$  is the Decision Variable denoting if a room is occupied. The Room  $r$  is used for Class  $c$  on Day  $d$  during shift  $t$
- Set  $A_{r,c}$  is the set of binaries denoting whether Room  $r$  has the ability to host Class  $c$ .
- Set  $B_{c,d,t}$  is the set of binaries denoting whether there is a Bargaining Unit in Class  $c$  on Day  $d$  during shift  $t$ .
- Set  $C$  is the set associated with index  $c$ , and list the courses.
- Set  $D$  is the set associated with index  $d$ , and list the working days for a fiscal year.
- Set  $F$  is a subset of Set  $D$  and contains only the days that a Bargaining Unit may not end on.
- Set  $M$  is a subset of set  $D$  and contains only the days that a Bargaining Unit may not begin on.
- Set  $N_c$  is the set of the number of instructors available for class  $c$ .
- Set  $P_c$  is the set of the number of radio pilot operators available for Class  $c$ .
- Set  $R_c$  is the set of the number of radio signal operators available for Class  $c$ .
- Set  $U_c$  is the set of durations of each Class  $c$ .

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, No bullets or numbering

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

## Objective Function

$$\text{Maximize } \sum_{e,d,t,r} X_{e,d,t,r}^-$$

The decision was made to make the decision variable focus on room availability because we believe that by maximizing the utilization of the training spaces, FAA can increase the number of students passing through the Academy.

## Constraints

$$\sum_e X_{e,d,t,r}^- \leq 1, \forall d, t, r$$

This constraint dictates that only one class may occur in a room at a given time on a given day.

$$\sum_e X_{e,d,t,r}^- = U_e, \forall e, t, r$$

This constraint dictates that if a room is assigned to a class, then that room must hold that class consecutively for its duration.

$$\sum_e E_e X_{e,d,t,r}^- \leq E_{total}, \forall d, t, r$$

This constraint dictates that the number of Evaluators assigned to every class during a given shift on a given day must not exceed the total number of instructors available.

$$\sum_e N_e X_{e,d,t,r}^- \leq N_{total}, \forall d, t, r$$

This constraint dictates that the number of Instructors assigned to every class during a given shift on a given day must not exceed the total Evaluators available.

$$\sum_e P_e X_{e,d,t,r}^- \leq P_{total}, \forall d, t, r$$

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt



This constraint dictates the number of Radio Signal Operators assigned to every class during a given shift on a given day must not exceed the total Radio Signal Operators available.

$$\sum_{\epsilon} R_{\epsilon} X_{c,d,t,r}^{-} \leq R_{total}$$

This constraint dictates that a class may not be assigned to a room that cannot support that class.

$$A_{r,\epsilon} \geq X_{c,d,t,r}, \forall c, d, t, r$$

This constraint dictates that Bargaining Unit classes may not begin on Mondays or days after holidays (contained in set M).

$$B_{c,d,t} X_{c,d,t,r} = 0, c = 1, d \in M, \forall t, r$$

This constraint dictates that Bargaining Unit classes may not end on Fridays or days before holidays (contained in set F).

$$B_{c,d,t} X_{c,d,t,r} = 0, c = last, d \in F, \forall t, r$$

This constraint dictates that classes must follow the order in Set C, and that there must not be gaps between classes, creating a unified block or complete course offering.

$$X_{c,d,t,r} = X_{c+1,d+1,t,r}, \forall c, d, t, r$$

This constraint dictates that classes must follow the order in Set C, and that there must not be gaps between classes, creating a unified block or complete course offering.

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Left, Space Before: 12 pt, After: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt

## 6. Recommendations and Benefits Technical Alternatives

After reviewing our original methodology, two optimization engines became the front runners for the model implementation: MATLAB and Gurobi. The model was coded into both engines and tested for performance. The results of that testing are listed in the following sections.

$$\min_x f^T x \text{ subject to } \begin{cases} x(\text{intcon}) \text{ are integers} \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub. \end{cases}$$

Figure 1: MIP Matrix Notation

### 6.1 MATLAB Model Composition

MATLAB is a scientific and mathematical software designed to easily build and manipulate matrices. Implementing the model in MATLAB required converting our optimization model into matrix notation, which can be seen in the figure below.

Integer programs are enacted in MATLAB using the INTLINPROG function, which is displayed below.

```
X = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub);
```

The f matrix contains the coefficients for the variables. For our model this is a 7,350,000 x 1 dimension matrix of negative ones (the coefficients are negative because MATLAB minimizes the model). The intcon matrix contains a row vector of the variable indices that are integer. This matrix contains a list of all integers between 1 and 7,350,000 because all of the variables in this problem are integers.

Formatted: Space After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Caption, Centered

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

The A and Aeq matrices contain the coefficients for the constraints. Each of these constraints is built using for-loops and subsets of each index. In our model, these constraints were ones or zeros, indicating that a particular variable was involved in a single constraint.

The b and beq matrices contain the values that the constraints must be smaller than. These matrices were created in a similar fashion to the A and Aeq matrices, but utilized double values, like the number of instructors needed.

The ub and lb matrices are the upper and lower bound matrices for the variables. Each is a 1 x 7,350,000 row vector, with every index corresponding to one value. Since these variables are binary, the upper bound matrix is composed entirely of ones while the lower bound matrix is composed entirely of zeros.

The creation of these matrices can be seen in the MATLAB code in Appendix A.

As a part of model development, we also developed code to read in data from matrices in separate .csv files. These files contained data pertaining to room ~~availability~~availability for each class and the number and type of staff needed for each class. The content of these .csv files can be seen in Appendix C.

## 6.2 MATLAB Model Performance

While MATLAB can quickly manipulate matrices of data, the software's optimization package can run at slow pace comparable to other commercial solvers. This was documented in Hans Mittlemen's comparison study of mixed integer programming solvers at Arizona State University, where Dr. Mittlemen documented that MATLAB could solve a MIP at 1/50<sup>th</sup> the speed of the other solvers. Dr. Mittlemen's study can be found at <http://plato.asu.edu/ftp/milpc.html>.

For our testing, we utilized a computer with 16GB of RAM with a quadcore, Intel i7 processor as well as a 32GB RAM, 12 core supercomputer at Oklahoma State University's High Process Computing Center. Each computer utilized MATLAB version 2016a.

In our testing, MATLAB's ~~these~~ limitations made themselves apparent. In our first test case, using the 16GB RAM computer, we reduced the model by a factor of 11,485 by reducing the number of variables from 7,350,000 to only 640. We loaded the A matrix completely with ones and attempted to run the model. However, ~~we were met~~received with the following error:

Formatted: Font: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Font: 12 pt

The screenshot shows the MATLAB workspace and command window. The workspace contains variables: A (2260x640 double), Aeq (72x640 double), b (2260x1 double), beq (72x1 double), f (640x1 double), and intcon (640x1 double). The command window shows the following code and error messages:

```

15 Staff_Needed = zeros(4,6,4); % (Type,Class,Section)
16
>> FAAModel_Small
Error using intlinprog (line 132)
INTLINPROG stopped because some objective or constraint matrix coefficients are too large in magnitude.

Error in FAAModel_Small (line 332)
X = intlinprog(f,intcon,A,b,Aeq,beq);

```

Figure 3: MATLAB Error

Figure 2: MATLAB Error

[This indicates that MATLAB, even at this small scale, exhibits scale issues with processing capabilities.](#)

[Our next test case involved populating the A matrix of the smaller model with the correct values. This allowed the model to run, but returned an optimal value of zero. This optimal value can easily be discredited by inspection, with hundreds of variables in the objective function to improve the optimal value. Further research will be needed required by the FAA to determine why the model would return such value.](#)

[Our final test case involved running the full-scale model with all 7,350,000 variables on the 32GB super computer. The model ran for 15 minutes \(to fully populate the large arrays\) but would error out after 15 minutes with the same error in Figure 1.](#)

[Our research and testing of the MATLAB model demonstrates that while MATLAB may have the potential to easily arrange data into matrices for processing, the actual optimization engine for mixed integer programming may hold some serious drawbacks.](#)

### 6.3 Gurobi Model Composition

[Gurobi is an industry leading optimization solver. In Dr. Mittlemen's study, it outperformed most commercial solvers in optimizing mixed integer programs. This is due to the engines state of the art, hybrid algorithms and custom methods that allow the user to quickly generate variables and constraints. Gurobi can be interfaced with many different programming languages, though we utilized Python for this implementation.](#)

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Caption, Centered, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, No bullets or numbering

Formatted: Normal, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, No bullets or numbering

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Heading 2, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Gurobi requires that the user provide several inputs to set up the model. The first is the model sense. For our case, this was a maximization model. Next we generated the variables using nested for-loops to properly index each variable. After that, we generated the constraints using for-loops and the custom Gurobi quicksum method, as shown below.

*Figure 3: Gurobi Quicksum function used to generate constraints*

```
for s in section:
    for c in clss:
        for d in day:
            for t in tme:
                for r in room:
                    m.addConstr((quicksum(x[s,c,o,d,t,r]
                                           for o in offering)) <=1)
```

*Figure 4: Gurobi Quicksum function used to generate constraints*

Gurobi will automatically generate constraints based on the given indices, and sum data sets using another for-loop.

After all the variables and constraints were added, the senior design team ordered the model to run. Gurobi began solving the program with its presolve feature, reducing the model in size and needed processing time. After, this Gurobi solved the optimization problem using its algorithm and returned an optimal objective value.

Full code for the Gurobi model can be found in Appendix B.

We also developed code to import data from the availability matrix (much like in the MATLAB model), but we also hardcoded some of the staff data into the model for ease of editing.

#### 6.4 Gurobi Model Performance

We tested the Gurobi model using the same computer with 16GB of RAM with a quadcore, Intel i7 processor we used to test the MATLAB model. We also executed the Python script in the Spyder integrated development environment. We utilized version 3.5 of Python and version 6.5 of Gurobi.

Our test case for this model, we ran the full model with 7,350,000 variables. It took the Python code just under 5 minutes to populate the data arrays and only 17 seconds to solve the model. The program returned an objective value of 13,102, which is the number of offerings in specific rooms at a particular time on a particular day for a given class in a given section. This number passes the eye test, and indicates, unlike the MATLAB model, the problem has a solution.

Formatted: Font: 12 pt

Formatted: Caption, Centered

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Caption, Centered, Space Before: 0 pt, After: 0 pt, Line spacing: single

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

The model utilized 12GB of RAM of the computer's available 16GB and processed in just over 5 minutes, indicating that this model can be successfully run on a business user's personal computer without the need to add additional RAM or processing power.

## 7. Recommendations Benefits

4. After performing the testing, the senior design, our team would like to make the following recommendations:

4.1 Recommendations We would highly recommends using the Gurobi implementation of the model over the MATLAB implementation for several reasons.

### 7.1 Benefits Time Commitment

As mentioned earlier, MATLAB can take up to 50 times longer than Gurobi to provide valuable scheduling information. In our testing, the MATLAB model (with the help of the massive processing power of a supercomputer) took 15 minutes to simply load the data in the large scale model before crashing INTLINPROG. Gurobi, on the other hand, took only 5 minutes to run the model in its entirety and provide information.

The business users' time is valuable, and formatting data to send to a supercomputer and wait for the results could burn through the time that managers are currently using on control tasks and potentially outnumber the amount of time currently spent making the schedule by hand.

4.2 Using the Gurobi model would reduce the current time investment from 50 total hours to only 4 hours (assuming each section manager runs the program 6 times and spends 30 minutes making edits to the section schedule).

4.3

### 7.2 Student Throughput

Both models have the potential to increase student throughput. The senior design team's last version of the Gurobi program produced an output that estimated an increase of 109 students.

Formatted: Space After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Font: 12 pt

Formatted: Font: 12 pt

Formatted: Normal, Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li, No bullets or numbering

Formatted: Font: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Line spacing: Multiple 1.15 li

Formatted: Space Before: 12 pt, After: 12 pt

Formatted: Space After: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt

Formatted: Space Before: 12 pt, After: 12 pt, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.27"

was not necessarily feasible. The program allocated an unrealistic schedule (placing little to no offerings of classes 2 and 3). However, it is important to note that the team still believes this program can improve the current scheduling practices at the FAA. The senior design team calculated that the FAA has room in their schedule for at most 180 more students per fiscal year. This number was calculated by looking at the utilization of the FAA's labs. While the most recent Gurobi program was unable to produce a feasible solution the team still believes that the program should still be able to find a schedule that would increase lab utilization and thus student throughput.

The last iteration of the Gurobi program and the math model it is based off of will require further analysis and edits to get to working condition. Should these edits be made however it is believed that the FAA can see the benefits mentioned. Further explanation of what analysis and edits will be necessary are discussed in the addendum. This increase is for one fiscal year meaning that the FAA can make up for their 400 controller deficit over the span of four fiscal years. According to the 2016 Air Traffic Controller Workforce plan, the FAA plans to make up for this deficit over the course of 10 years. Not only does this increase in maximum student throughput solve the FAA's current problem, but it helps reduce the risk of future ones as well.

The current deficit was caused due to an unforeseen inability meet the planned quota. The FAA's ability to recover from these moments depends entirely upon the maximum student throughput allowed. By increasing this number the FAA can better prepare themselves for years like 2015.

Formatted: Font: 12 pt

## Appendix A: MATLAB Model

```
%For the purposes of Testing, Values are assigned here.  
C0 = 4;  
C1 = 5;  
D0 = 5;  
D1 = 250;  
O0 = 2;  
O1 = 35;  
R0 = 2;  
R1 = 21;  
E0 = 4*C0*O0*D0*2*R0; %640  
E1 = 4*C1*O1*D1*2*R1; %7350000  
  
%Import Staff Data and Room Ability Matrices  
Avail = zeros(R0,C0,4); %(Room, Class, Section)  
Staff Needed = zeros(4,6,4); %(Type,Class,Section)  
  
Avail(1:end,1:end,1) = csvread('BasicRoomClassMatrix.csv',1,1,[1,1,R0,C0]);  
Avail(1:end,1:end,2) = csvread('TowerRoomClassMatrix.csv',1,1,[1,1,R0,C0]);  
Avail(1:end,1:end,3) = csvread('RTFRoomClassMatrix.csv',1,1,[1,1,R0,C0]);  
Avail(1:end,1:end,4) = csvread('EnRouteRoomClassMatrix.csv',1,1,[1,1,R0,C0]);
```

```

Staff_Needed(1:end,1:end,1) = csvread('BasicInstructorClassMatrix.csv',1,1);
Staff_Needed(1:end,1:end,2) = csvread('TowerInstructorClassMatrix.csv',1,1);
Staff_Needed(1:end,1:end,3) = csvread('RTFInstructorClassMatrix.csv',1,1);
Staff_Needed(1:end,1:end,4) =
csvread('EnRouteInstructorClassMatrix.csv',1,1);

%Start Timer
tic;

% Create f matrix
f = -1*ones(4,C0,O0,D0,2,R0);
for c = 1:C0-1
    for o = 1:O0
        for d = 1:D0
            for t = 1:2
                for r = 1:R0
                    f(1,c,o,d,t,r) = 0;
                end
            end
        end
    end
end
f = reshape(f,[E0,1]);

% Create intcon matrix
intcon = zeros(E0,1);
for i = 1:E0
    intcon(i) = i;
end

% Create A Matrix
ABeg = zeros(4,4,O0,D0,2,R0);

% CONSTRAINT 1: Number of Evaluators in Basics and En Route
rows = 2*4*D0*2; %This is the product of the for-loop indices
A1 = zeros(rows,E0);
b1 = zeros(rows,1);
row_count = 1;
for s = [1,4]
    for c = 1:C0
        for d = 1:D0
            for t = 1:2
                AMid = ABeg;
                AMid(s,c,1:end,d,t,1:end) = ones(O0,R0);
                AMid = reshape(AMid,[1,E0]);
                A1(row_count,:) = AMid;
                if Staff_Needed(4,c,s) ~= 0
                    b1(row_count,1) =
Staff_Needed(4,6,s)/Staff_Needed(4,c,s); %Total/Needed
                else
                    b1(row_count,1) = 0;
                end
                row_count = row_count + 1;
            end
        end
    end
end
end
end

```



```

end

% CONSTRAINT 2: Number of Evaluators in Tower and RTF
rows = 4*D0*2; %This is the product of the for-loop indices
A2 = zeros(rows,E0);
b2 = zeros(rows,1);
row count = 1;
for c = 1:C0
    for d = 1:D0
        for t = 1:2
            AMid = ABeg;
            AMid(s,c,1:end,d,t,1:end) = ones(O0,R0);
            AMid = reshape(AMid,[1,E0]);
            A2(row count,:) = AMid;
            if Staff Needed(4,c,s) ~= 0
                b2(row count,1) =
Staff Needed(4,6,s)/Staff Needed(4,c,s); %Total/Needed
            else
                b2(row count,1) = 0;
            end
            row count = row count + 1;
        end
    end
end

% CONSTRAINT 3: Number of RSOs
rows = 4*4*D0*2; %This is the product of the for-loop indices
A3 = zeros(rows,E0);
b3 = zeros(rows,1);
row count = 1;
for s = 1:4
    for c = 1:C0
        for d = 1:D0
            for t = 1:2
                AMid = ABeg; %Reset AMid
                AMid(s,c,1:end,d,t,1:end) = ones(O0,R0);
                AMid = reshape(AMid,[1,E0]);
                A3(row count,:) = AMid;
                if Staff Needed(3,c,s) ~= 0
                    b3(row count,1) =
Staff Needed(3,6,s)/Staff Needed(3,c,s); %Total/Needed
                else
                    b3(row count,1) = 0;
                end
                row count = row count + 1;
            end
        end
    end
end

% CONSTRAINT 4: Number of RPOs
rows = 4*4*D0*2;
A4 = zeros(rows,E0);
b4 = zeros(rows,1);
row count = 1;
for s = 1:4

```

```

    for c = 1:C0
        for d = 1:D0
            for t = 1:2
                AMid = ABeg; %Reset AMid
                AMid(s,c,1:end,d,t,1:end) = ones(O0,R0);
                AMid = reshape(AMid,[1,E0]);
                A4(row_count,:) = AMid;
                if Staff_Needed(2,c,s) ~= 0
                    b4(row_count,1) =
Staff_Needed(2,6,s)/Staff_Needed(2,c,s); %Total/Needed
                else
                    b4(row_count,1) = 0;
                end
                row_count = row_count + 1;
            end
        end
    end
end

% CONSTRAINT 5: Number of Instructors
rows = 4*4*D0*2;
A5 = zeros(rows,E0);
b5 = zeros(rows,1);
row_count = 1;
for s = 1:4
    for c = 1:4
        for d = 1:D0
            for t = 1:2
                AMid = ABeg; %Reset AMid
                AMid(s,c,1:end,d,t,1:end) = ones(O0,R0);
                AMid = reshape(AMid,[1,E0]);
                A5(row_count,:) = AMid;
                if Staff_Needed(1,c,s) ~= 0
                    b5(row_count,1) =
Staff_Needed(1,6,s)/Staff_Needed(1,c,s); %Total/Needed
                else
                    b5(row_count,1) = 0 ;
                end
                row_count = row_count + 1;
            end
        end
    end
end
end

% CONSTRAINT 6: Classes Must Occur in Order
Duration = 1; %This input will be changed to accomdate GUI later.
rows = 3*3*O0*D0;
A6 = zeros(rows,E0);
row_count = 1;
for s = 1:3
    for c = 1:C0-1
        for o = 1:O0
            for d = 1:D0
                for t = 1:2
                    for r = 1:R0
                        AMid = ABeg; %Reset AMid

```

```

                if (d+Duration) < D0
                    AMid(s,c+1,o,d+Duration,1:end,1:end) = -
1*ones(2,R0);
                end
                AMid(s,c,o,d,t,r) = 1;
                AMid = reshape(AMid,[1,E0]);
                A6(row_count,:) = AMid;
                row_count = row_count + 1;
            end
        end
    end
end
end
end
end
b6 = zeros(3*3*O0*D0*2*R0,1);

% CONSTRAINT 7: Each room can only host certain classes.
rows = E0;
A7 = zeros(rows,E0);
b7 = zeros(rows,1);
row_count = 1;
for s = 1:4
    for c = 1:C0
        for o = 1:O0
            for d = 1:D0
                for t = 1:2
                    for r = 1:R0
                        AMid = ABeg; %Reset AMid
                        AMid(s,c,o,d,t,r) = 1;
                        AMid = reshape(AMid,[1,E0]);
                        A7(row_count,:) = AMid;
                        b7(row_count,1) = Avail(r,c,s);
                        row_count = row_count + 1;
                    end
                end
            end
        end
    end
end
end
end
end

% CONSTRAINT 8: Basic Classes must be followed by En Route or Tower
rows = O0*D0*2;
A8 = zeros(rows,E0);
row_count = 1;
for o = 1:O0
    for d = 1:D0
        for t = 1:2
            AMid = ABeg; %Reset AMid
            AMid(2,1,o,d,t,1:end) = -1*ones(R0,1);
            AMid(4,1,o,d,t,1:end) = -1*ones(R0,1);
            AMid(1,4,o,d,t,r) = 1;
            AMid = reshape(AMid,[1,E0]);
            A8(row_count,:) = AMid;
            row_count = row_count + 1;
        end
    end
end
end
end
end

```

```

end
b8 = zeros(rows,1);

% CONSTRAINT 9: Now two offerings of a particular section can be held in
% the same room at the same time.
rows = E0;
A9 = zeros(rows,E0);
row count = 1;
for s = 1:4
    for c = 1:C0
        for o = 1:O0
            for d = 1:D0
                for t = 1:2
                    for r = 1:R0
                        %Prime Sets
                        s_prime = 1:4;
                        s_prime(s) = [];
                        c_prime = 1:4;
                        c_prime(c) = [];
                        d_prime = 1:D0;
                        d_prime(d) = [];
                        o_prime = 1:O0;
                        o_prime(o) = [];
                        %Populate Arrays
                        AMid = ABeg; %Reset AMid
                        AMid(s,c,o,d,t,r) = 1;
                        AMid(s_prime,c_prime,o_prime,d_prime,t,r) =
ones(3,3,O0-1,D0-1);
                        AMid = reshape(AMid,[1,E0]);
                        A9(row count,:) = AMid;
                        row count = row count + 1;
                    end
                end
            end
        end
    end
end
b9 = ones(rows,1);

% CONSTRAINT 10: Bad Bargaining Unit Start Days
Bad Start = [1];
BU Offerings = [1];
rows = 4*numel(BU Offerings)*numel(Bad Start)*2*R0;
A10 = zeros(rows,E0);
row count = 1;
for c = 1:C0
    for o = BU Offerings
        for d = Bad Start
            for t = 1:2
                for r = 1:R0
                    AMid = ABeg; %Reset AMid
                    AMid(3,c,o,d,t,r) = 1;
                    AMid = reshape(AMid,[1,E0]);
                    A10(row count,:) = AMid;
                    row count = row count + 1;
                end
            end
        end
    end
end

```

```

end
end
end
end
b10 = zeros(rows,1);

% CONSTRAINT 11: Bad Bargaining Unit Start Days
Bad End = [5];
rows = 4*numel(BU Offerings)*numel(Bad End)*2*R0;
A11 = zeros(rows,E0);
row count = 1;
for c = 1:C0
    for o = BU Offerings
        for d = Bad End
            for t = 1:2
                for r = 1:R0
                    AMid = ABeg; %Reset AMid
                    AMid(3,c,o,d,t,r) = 1;
                    AMid = reshape(AMid,[1,E0]);
                    A11(row count,:) = AMid;
                    row count = row count + 1;
                end
            end
        end
    end
end
b11 = zeros(rows,1);

% CONSTRAINT 12: Bad Bargaining Unit Shifts - No Nights
rows = 4*numel(BU Offerings)*D0*R0;
A12 = zeros(rows,E0);
row count = 1;
for c = 1:C0
    for o = BU Offerings
        for d = D0
            for r = 1:R0
                AMid = ABeg; %Reset AMid
                AMid(3,c,o,d,1,r) = 1;
                AMid = reshape(AMid,[1,E0]);
                A12(row count,:) = AMid;
                row count = row count + 1;
            end
        end
    end
end
b12 = zeros(rows,1);

% CONCAT Matrix A
A = vertcat(A1,A2,A3,A4,A5,A6,A7,A8,A9);

%CONCAT Matrix b
b = vertcat(b1,b2,b3,b4,b5,b6,b7,b8,b9);

% CONCAT Matrix Aeg
Aeg = vertcat(A10,A11,A12);

```

```
%CONCAT Matrix beg  
beg = vertcat(b10,b11,b12);  
  
%Lower Bound  
lb = zeros(E0,1);  
  
%Upper Bound  
ub = ones(E0,1);  
  
% Run Model  
X = intlinprog(f,intcon,A,b,Aeq,beg,lb,ub);  
  
toc;  
disp('Done')
```

## Appendix B: Gurobi Model

```
# -*- coding: utf-8 -*-  
"""  
Created on Sat Mar 25 10:06:21 2017  
  
@author: james  
"""  
  
from gurobipy import *  
import time  
import csv  
import numpy as np  
from numpy import genfromtxt  
import re  
  
#Start Clock  
start = time.time()  
output = []  
  
#Create Model  
m = Model("mip1")  
m.setParam('TimeLimit', 300)  
  
#Load Index Data into Lists  
section = [1,2,3,4]  
class = [1,2,3,4,5]
```

```

offering = []
for i in range(1,36):
    offering.append(i)
day = []
for i in range(1,282):
    day.append(i)
tme = [0,1]
room = []
for i in range(1,21):
    room.append(i)

#Load Parameter Data into Lists
Amid = genfromtxt('BasicRoomClassMatrix.csv', delimiter=',')
A1 = Amid[1:,1:]
Amid = genfromtxt('TowerRoomClassMatrix.csv', delimiter=',')
A2 = Amid[1:,1:]
Amid = genfromtxt('RTFRoomClassMatrix.csv', delimiter=',')
A3 = Amid[1:,1:]
Amid = genfromtxt('EnRouteRoomClassMatrix.csv', delimiter=',')
A4 = Amid[1:,1:]
A = np.vstack((A1,A2,A3,A4))
A = A.reshape(4,21,5)

durations = [
    [25,0,0,0,0],
    [13,6,1,14,3],
    [8,19,2,0,0],
    [18,11,12,16,2]
]

nextFiscalDays = []
for i in range(251,282):

```



```
nextFiscalDays.append(i)

dates badStart = []
for i in range(1,26):
    dates badStart.append(i*10)

offerings bu = [1,12,14,22,29]

instructors = [
    [2,0,0,0,0],
    [2,2,2,2,1],
    [1,2,4,0,0],
    [2,10,4,14,1]
]
instructors max = [130,130,130,130]

RSOs = [
    [0,0,0,0,0],
    [0,6,0,8,4],
    [2,16,16,0,0],
    [0,0,0,18,12]
]
RSOs max = [0,0,0,50]

RPOs = [
    [0,0,0,0,0],
    [0,11,4,8,1],
    [0,24,24,0,0],
    [0,0,0,6,6]
]
RPOs max = [0,50,50,50]
```

```

evaluator = [
    [0,0,0,0,1],
    [0,0,0,1,1],
    [0,0,1,0,0],
    [0,0,0,0,6]
]
evaluator_max = [7,7,7,7]

#Create Variables
x={}
for s in section:
    for c in class:
        for o in offering:
            for d in day:
                for t in time:
                    for r in room:
                        x[s,c,o,d,t,r] = m.addVar(obj=1,
vtype=GRB.CONTINUOUS,
name='x[{} {} {} {} {} {}]'.format(s,c,o,d,t,r),
ub=1, lb=0)
m.ModelSense = GRB.MAXIMIZE
m.update()

#Add Constraints
try:
    #No two classes in the same room at the same time
    for s in section:
        for c in class:
            for d in day:
                for t in time:
                    for r in room:
                        m.addConstr((quicksum(x[s,c,o,d,t,r]

```

```

_____ for o in offering)) <=1)
_____

#Must not exceed max number of instructors
_____ for s in section:
_____ for c in class:
_____ for d in day:
_____ for t in tme:
_____ for r in room:
_____ m.addConstr((instructors[s-1][c-
_____ 1]*quicksum(x[s,c,o,d,t,r]
_____ for o in offering))
_____ <= instructors max[s-1])
_____

#Must not exceed max number of RSOs
_____ for s in section:
_____ for c in class:
_____ for d in day:
_____ for t in tme:
_____ for r in room:
_____ m.addConstr((RSOs[s-1][c-1]*quicksum(x[s,c,o,d,t,r]
_____ for o in offering))
_____ <= RSOs max[s-1])
_____

#Must not exceed max number of RPOs
_____ for s in section:
_____ for c in class:
_____ for d in day:
_____ for t in tme:
_____ for r in room:
_____ m.addConstr((RPOs[s-1][c-1]*quicksum(x[s,c,o,d,t,r]
_____ for o in offering))
_____ <= RPOs max[s-1])
_____

```

```

#Must not exceed max number of Evaluators for Sections 1 and 4
sprime = [1,4]
for s in sprime:
    for c in class:
        for d in day:
            for t in tme:
                for r in room:
                    m.addConstr((evaluators[s-1][c-
1]*quicksum(x[s,c,o,d,t,r]
                                for o in offering))
                                <= evaluators_max[s-1])

#Must not exceed max number of shared Evaluators for Sections 2 and 3
sprime = [2,3]
for s in sprime:
    for c in class:
        for d in day:
            for t in tme:
                for r in room:
                    m.addConstr((evaluators[s-1][c-
1]*quicksum(x[s,c,o,d,t,r]
                                for s in sprime
                                for o in offering))
                                <= evaluators_max[s-1])

#Prevents classes from starting after fiscal year
for s in section:
    for c in class:
        for o in offering:
            for d in nextFiscalDays:
                for t in tme:
                    for r in room:

```

```

m.addConstr(x[s,c,o,d,t,r] == 0)

#Classes in a section must be consecutive
cprime = [1,2,3,4]
day = []
for i in range(1,251):
    day.append(i)
for s in section:
    for c in cprime:
        for d in day:
            for t in tme:
                for o in offering:
                    for r in room:
                        m.addConstr(x[s,c,o,d,t,r] <=
                                    (quicksum(x[s,c+1,o,d+durations[s-1][c-
1],tprime,rprime]
                                             for rprime in room
                                             for tprime in tme)))

#Every Basics Section Must be Followed by either En Route or Tower
for c in clss:
    for d in day:
        for o in offering:
            for t in tme:
                m.addConstr(x[1,c,o,d,t,r] ==
                            (quicksum(x[2,1,o,d,t,r]
                                       for r in room)) +
                            (quicksum(x[4,1,o,d,t,r]
                                       for r in room)))

#A Bargaining Unit may not begin on certain days
for c in clss:

```

```

    for d in dates badStart:
        for o in offerings bu:
            for t in tme:
                for r in room:
                    m.addConstr(x[3,c,o,d,t,r] == 0)

#A Bargaining Unit may not end on certain days
for c in cls:
    for d in dates badStart:
        for o in offerings bu:
            for t in tme:
                for r in room:
                    m.addConstr(x[3,c,o,d+durations[s-1][c-1],t,r] == 0)

#Certian rooms can only host certain classes
for s in section:
    for c in cls:
        for o in offering:
            for d in day:
                for t in tme:
                    for r in room:
                        m.addConstr(x[s,c,o,d,t,r] <= A[s-1,r-1,c-1])

#Ensure Minimums
for s in section:
    m.addConstr(18*quicksum(x[s,1,o,d,t,r]
                            for o in offering
                            for d in day
                            for t in tme
                            for r in room) >= 500)

#Ensure Minimums

```

```

    for s in section:
        for c in cprime:
            for o in offering:
                for d in day:
                    for t in tme:
                        m.addConstr(quicksum(x[s,c,o,d,t,r]
                                                for r in room) ==
                                    (quicksum(x[s,c+1,o,d+durations[s-1][c-
1],tprime,rprime]
                                                for rprime in room
                                                for tprime in tme)))

    m.update()

    #Solve
    start_solve =time.time()
    m.optimize()
    end_solve = time.time()

    #Print Results
    if m.status==GRB.OPTIMAL:
        print("Solve Time: " + str(end_solve-start_solve))
    #    gudStuff = m.getVars()
    #    for i in range(0,len(gudStuff)):
    #        gudStuff[i] = str(gudStuff[i])
    #        subs = re.search('\[(.+?)]', gudStuff[i])
    #        values = re.search('value(.+?)\)', gudStuff[i])
    #        if subs:
    #            output.append([subs.group(1),float(values.group(1))])
    m.printAttr('x')

```

```
else:
    print("Not Optimal")

except GurobiError:
    print("Error occurred")

#End Clock
print (str(time.time() - start) + " Seconds Total")
```



## Appendix C: Input Tables (.csv)

### Basics Staff Matrix

	Academics					Max
Instructors	2	0	0	0	0	130
RPOS	0	0	0	0	0	100
RSOs	0	0	0	0	0	50
Evaluators	0	0	0	0	1	7

### Tower Staff Matrix

	Academics	TT/3D	C+	Lab	PA	Max
Instructors	2	2	2	2	1	130
RPOS	0	11	4	8	1	100
RSOs	0	6	0	8	4	50
Evaluators	0	0	0	1	1	7

Formatted: Font: (Default) +Body (Calibri), 11 pt, Not Bold

Formatted: Font: (Default) +Body (Calibri), 11 pt, Not Bold

Formatted: Font: (Default) +Body (Calibri), 11 pt, Not Bold

Formatted: Font: (Default) +Body (Calibri), 11 pt, Not Bold

Formatted: Font: (Default) +Body (Calibri), 11 pt, Not Bold

### Tower Staff Matrix

	Academics	Lab	Personal Assessment		Max
Instructors	1	2	4	0	130
RPOS	0	24	24	0	100
RSOs	2	16	16	0	50
Evaluators	0	0	1	0	7

### En Route Staff Matrix

	Non-Radar Academics	Non-Radar Lab	ERAM Academics	Eram Lab	Evaluation	Max
Instructors	2	10	4.4	14	1	130
RPOS	0	0	0	6	6	100
RSOs	0	0	0	18	12	50
Evaluators	0	0	0	0	6	7

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

### Basics Room Matrix

	Academics				
RTF 207	1	0	0	0	0

Formatted: Font: Not Bold

<a href="#">RTF 204</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">AT 147</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">AT 157</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">AT 158</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">ACAD B5</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">ACAD B25</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

#### Tower Room Matrix

	<a href="#">Academics</a>	<a href="#">IT/3D</a>	<a href="#">C+</a>	<a href="#">Lab</a>	<a href="#">PA</a>
<a href="#">152/145</a>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">160/24</a>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">161/163</a>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">Tss 1-6</a>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>
<a href="#">Tss 7-12</a>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>

#### RTF Room Matrix

	<a href="#">Academics</a>	<a href="#">Lab</a>	<a href="#">Personal Assessment</a>		
<a href="#">104</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">107</a>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<a href="#">103</a>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">114</a>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">115</a>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">117</a>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>

#### En Route Room Matrix

	<a href="#">Non-Radar Academics</a>	<a href="#">Non-Radar Lab</a>	<a href="#">ERAM Academics</a>	<a href="#">Eram Lab</a>	<a href="#">Evaluation</a>
<a href="#">165</a>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">168</a>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">213c</a>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">213D</a>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">Lab A</a>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>
<a href="#">Lab B</a>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>
<a href="#">159(12Person)</a>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
<a href="#">LabC(12Person)</a>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>

## Acknowledgements

The senior design senior design team would like to thank the following people for their help and contribution:

- Mr. Allen Glenn for his constructive criticism, feedback, time, and mentoring as the senior design team's Instructor of Record
- Dr. Farzad Yousefian for his feedback, time, and mentoring as the senior design team's Faculty Mentor
- Mr. Bill Dueease for his feedback, time, and mentoring as the senior design team's Industrial Advisor Board Advisor Coach.
- Dr. Nina Barker for her feedback, time and support as the senior design team's project Point of Contact
- Dr. Buchanan for his feedback, time, criticism, and support
- Dr. Baski for his feedback, time, criticism, and support