Air Force Operations Center Scheduling (AFOCS)

OR 680 / SYST 798
Capstone Project Final Report

Prepared for: Dr. Kathryn Laskey

Prepared by: Scott Genberg

Rebecca McCrabb

Ashley Rock

Amar Zabarah

Sponsors: Brian Collins (Kepler Research, Inc)

Matt Prebble (Kepler Research, Inc)

02 May 2011

[This page intentionally left blank.]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

[This page intentionally left blank.]

1. Executive Summary

U.S. Air Force operations require staffing several operation centers with trained and certified personnel. Scheduling the staffing of these operation centers is currently a time consuming manual process. The schedules must be prepared for the operation centers, training events, training resources, and trainers necessary to maintain current certification. The goal of this project is to develop an optimization scheduling model that will enable the automation of the scheduling tasks, currently performed manually, for a USAF group-level organization.

The project team developed an optimization model that provides for the automatic scheduling of fifteen operation centers. The optimization model also schedules required training, training resources and instructors. The model is capable of handling daily changes and producing a re-optimized solution while still adhering to all objective and constraints.

The emphasis of the project was to develop, test and analyze an optimization algorithm to improve efficiency and performance over the existing manual scheduling process. The scope of the project was limited, and no user interface or software code to interface with the current scheduling tool, TimePiece, was developed.

Design and development of the optimization algorithm, the model, was conducted through an iterative approach. The model was developed using Linear Programming and Integer Programming techniques. Team AFOCS used AIMMS v3.11 as the primary software tool.

The optimization model produces monthly schedules for personnel in four phases. The first phase focuses on scheduling the shifts for the operation centers, as fulfilling shift demand is the first priority. After shifts are scheduled, the model's second phase focuses on scheduling training events for each person to meet the monthly training requirements. The training events require personnel being trained, instructors to perform the training, and the necessary training resources, with each component having limited availability and potential scheduling conflicts. The third phase of the model entails scheduling the backup crew for each day. The fourth phase performs the assignment of students, instructors, and evaluators to the simulator training and evaluation events. After creation of the monthly original schedule for personnel, the last portion of the model development was to create the capability to reschedule mid-month due to an absence of one or more personnel.

The optimization algorithm was tested iteratively throughout the model development. The automated model provides a solution after about three hours of runtime. The model performs in much less time than the current scheduling process, which is approximately 1.5 weeks. Not only does the model reduce the scheduling process time down significantly, it also finds a solution within 2% of the relaxed linear programming solution's optimality.

2. Problem Definition

2.1 Introduction

U.S. Air Force operations require staffing numerous operation centers (op centers) with trained and certified personnel. Scheduling the staffing of these op centers is a time consuming manual process. Scheduling includes not only staffing the op centers, but also scheduling the training events, training resources, and trainers necessary to maintain current certification. The goal is to develop an optimization scheduling model that will enable the automation of the scheduling tasks for a USAF group-level organization (Group).

2.2 Background

The Group provides staffing to 15 op centers. Each op center requires two functional positions: Crew Commander (CDR) and Deputy Crew Commander (DEP). The shifts are 24 hours long (7 a.m. to 7 a.m.). Shifts are referred to as "Alerts," however, in this document the term shift will be used.

2.3 Organization

The USAF group consists of three squadrons. Each squadron is responsible for staffing five op centers, one of which is the Squadron Command Post (SCP) which requires an additional certification for the personnel manning this op center.

2.4 Personnel

Personnel are categorized according to their functional roles. The four functional roles are:

1. Crew member (CW)
2. Instructor (INST)
3. Evaluator (EVAL)
4. Flight Commander (FLT CDR)

CWs are the individuals whose primary function is to perform op center duty. Each shift requires one CDR and one DEP for each op center. Pairing the same CDR with the same DEP is referred to as "crew integrity". Maintaining a crew integrity level of 80%, that is 80% of the time each person is assigned to a shift or training they are paired with the same partner, is a high priority for the organization. CW can pull up to a maximum of 8 shifts in a calendar month. All CDRs can pull duty as DEPs if necessary.

The other three sets of individuals, in addition to their role as INST, EVAL, or FLT CDR, are also qualified as either CDR or DEP. These individuals are required to pull no more than two shifts per calendar month.

2.5 Required Training

Individuals are required to participate in mandatory monthly training events to maintain current certifications to perform duty. The monthly training events are:

- TR
- T1
- T3
- T4

Additionally, there is an annual evaluation (AE) requirement. Table 1 summarizes the training requirements for the personnel to be qualified to perform op center duty.

Table 1: Summary of Required Training

Training Event	Frequency	Type	Duration	Trainer	Comments
TR	Monthly	Simulator	4 hr	INST – 2 ea	
T1	Monthly	Classroom	8 hr	INST – 2 ea	
T3	Monthly	Classroom	4 hr	INST – 2 ea	Often paired with T4
T4	Monthly	Classroom	4 hr	INST – 2 ea	Often paired with T3
AE	Annually	Simulator	4 hr	EVAL – 3 ea	

There is a maximum of 45 calendar days between performing duty in the op center. However, for the personnel assigned to perform SCP duty, the maximum time limit is increased to 60 days between performing duty in the SCP.

2.6 Down Time

The day immediately following a shift is an off day ("O-day") for the personnel who performed duty during the shift. Additionally, all personnel are required to have nothing scheduled (training) during the twelve hours immediately preceding the shift, and must have at least 3 days between shifts. Sundays and federal holidays are normally training holidays with no scheduled training, however shifts are still scheduled.

2.7 Training Resources

The group has two simulators and numerous classrooms available to conduct the monthly required training. Simulator availability is dependent upon the operations tempo (op temp) for the group. The simulators are available in four 4-hour blocks from 7 a.m. to 11 p.m. during regular op tempo or in five 4-hours blocks from 6 a.m. to 2 a.m during increased op tempo. Simulator availability and classroom availability will each be constraints as the number of simulators and simulator slot times are limited as well as the number of available seats in each classroom.

3. Statement of Need

An optimization model needs to be developed for automatic scheduling of op centers to improve efficiency and performance of the existing scheduling process. Additionally, the optimization model must handle daily changes in personnel availability during the month and adjust the schedule to meet shift and training requirements. As a result, the primary objective of the AFOCS team is to develop a crew scheduling algorithm to assign personnel to shifts and required training for the desired month. To address the daily changes in crew availability, the team must also develop a dynamic algorithm to handle daily changes and produce a re-optimized solution while still adhering to all the scheduling conditions.

This algorithm was designed within the 14-weeks class period. The analysis of the concept and design of this algorithm was concluded by 6 May 2011. Also, a cumulative final report was prepared by 2 May 2011.

4. Scope

4.1 In Scope

The emphasis of the AFOCS team is to develop, test and analyze an optimization algorithm to improve efficiency and performance of the existing scheduling process.

4.2 Out of Scope

The AFOCS team is not responsible for implementing code to merge the algorithm with the current system.

5. Technical Approach

A standard Systems Engineering vee-design approach was applied to this project. Due to the very short period of performance, some of the activities occurred concurrently. The project was completed in five overlapping phases. The phases were:

- Requirements Development
- Research and Analysis
- Algorithm Development
- Test and Evaluation
- Delivery

Figure 1 shows the customized Systems Engineering approach to this project.

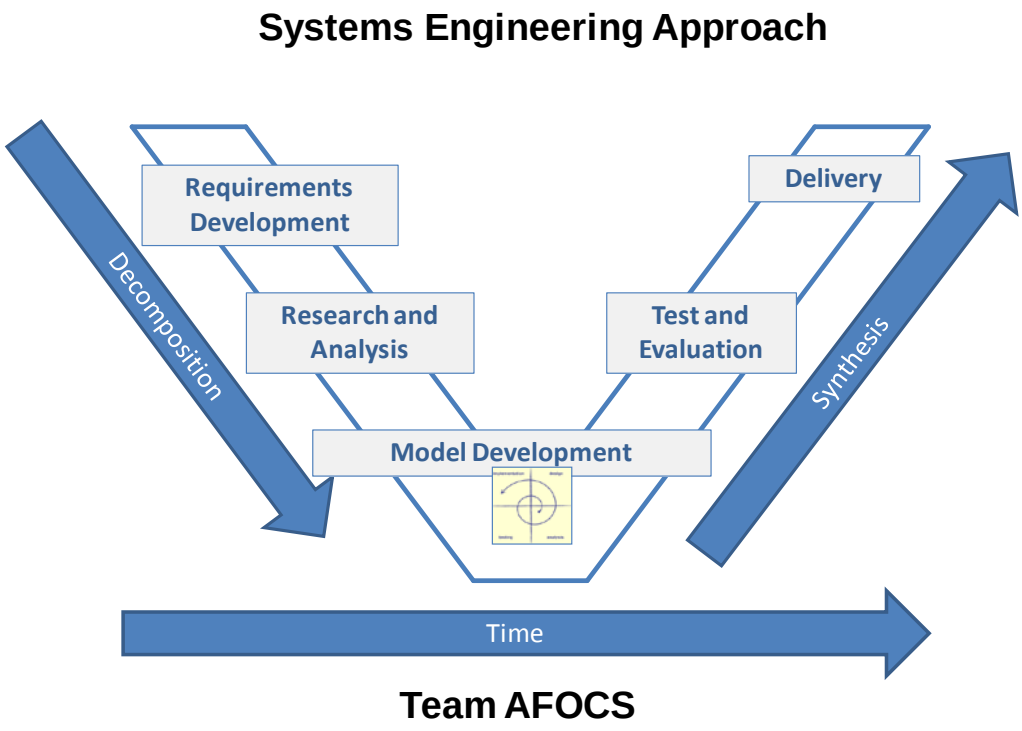


Figure 1: Team AFOCS Technical Approach

5.1 Requirements Development

Initially, Team AFOCS examined the sponsor's stated requirements, needs, and desired outcome. The team derived further detailed requirements. A large portion of the requirements phase was spent developing the functional requirements of the model. The sponsor provided feedback during the requirements development phase and approved the finalized functional requirements.

5.2 Research and Analysis

The Research and Analysis phase of the project consisted of two major interrelated activities. The first activity was the continuation of the requirements analysis from the first phase in the project to discover any previously unknown interrelations and dependencies.

The second major activity in this phase was a literature review for similar models and algorithms. The team used the resources of the George Mason University Systems Engineering and Operations Research department, in addition to any other corporate resources to which the individual team members might have access. TimePiece, a product developed by Kepler Research, Inc. is the current scheduling tool used by the USAF group. The current practices were also analyzed.

5.3 Algorithm Development

Design and development of the optimization algorithm, the model, was conducted through an iterative approach. The model was developed using Linear Programming and Integer

Programming techniques. And to satisfy the technical requirements team AFOCS used AIMMS v3.11 as the primary software tool (www.aimms.com).

The optimization produces monthly schedules for personnel in four phases. The first phase focuses on scheduling the shifts for the op centers, as fulfilling shift demand is the top priority. After shifts are scheduled, the model's second phase focuses on scheduling training events for each person to meet the monthly training requirements. The training events require the coordination of (1) personnel to complete the training, (2) INSTs to perform the training, and (3) simulators or classrooms to serve as the platform for training, with each component having limited availability and scheduling conflicts. The third phase entails scheduling the backup crew for each day, which consists of one CDR and one DEP who are not scheduled for shift or training on that day. The fourth phase performs the assignment of students, INSTs, and EVALs to simulator slots for the TR and AE simulation events.

After creation of the monthly original schedule for personnel, the last portion of the model development was to create the capability to reschedule mid-month due to an absence of one or more personnel. The model has two different methods to address personnel unavailability mid-month: a heuristic approach for 1-2 personnel becoming unavailable and a dynamic approach for more than 2 people becoming unavailable on a single problem day. The heuristic approach utilizes the backup crew to fill-in for a person who becomes unavailable, and then schedules another person for the backup crew to fill-in for the backup CW now assigned to a shift. The dynamic portion of the model modifies the schedule from the problem day to the end of the month and reassigns personnel to shifts and training to meet demand while minimizing deviation from the original schedule.

5.4 Test and Evaluation

The optimization algorithm was tested iteratively throughout the model development. Final testing was accomplished with simulated data. Testing and Evaluation are further described in section 8 Testing and Validation.

5.5 Delivery

The model will be delivered to the sponsor, Kepler Research, Inc. during the week of 2 May. Delivery will include a softcopy of the model in AIMMS and a softcopy of the model documentation.

6. Requirements

This section lists the project's requirements. As mentioned in Section 5.1 Requirements Development, the team met with the sponsor at the initial phase of the project to extract top level requirements, and continued to meet throughout the project phases to verify the requirements language.

6.1 System Definition

6.1.1 System Description

The AFOCS model is an optimization model that automates the current manually intensive scheduling process that is performed monthly.

6.1.2 System Users

The system users are the schedulers at the squadron and group level.

6.2 System Requirements

6.2.1 Functional Requirements (Op Center Staffing)

6.2.1.1 The model shall schedule 15 op centers plus one standby crew.

6.2.1.2 The model shall schedule each op center to be staffed by one Crew Commander and one Deputy Crew Commander.

6.2.1.3 The model shall schedule 24 hour shifts (7 a.m. to 7 a.m.) for each op center.

6.2.1.4 The model shall schedule an off day (O Day) after each alert.

6.2.1.5 The model shall schedule the Crew Commander positions with qualified crew members.

6.2.1.6 The model shall schedule the Deputy Crew Commander positions with qualified crew members. Personnel qualified as Crew Commander are able to staff the Deputy Crew Commander position.

6.2.1.7 The model shall verify that each crew member have completed required mandatory training in the previous calendar month. See section 6.2.2Functional Requirements (Mandatory Training).

6.2.1.8 Each squadron has one Squadron Command Post (SCP). The model shall schedule the staffing of the SCP by SCP qualified crew members. SCP qualification is an additional qualification beyond the Crew Commander/Deputy Crew Commander position.

6.2.1.9 The model shall verify that SCP qualified crew members accomplish one SCP alert at a minimum of every 60 days to maintain certification.

6.2.1.10 The model shall schedule each crew member at minimum of one alert every 45 days to maintain certification.

6.2.1.11 The model shall schedule Instructors, Evaluators, and Flight Commanders for at most two alerts per calendar month.

6.2.1.12 The model shall schedule Crew Members for no more than eight alerts per calendar month.

6.2.1.13 Pairing the same CDR with the same DEP is referred to as “crew integrity”; one of the organization’s goals is to maximize crew integrity. The model shall schedule crew members paired together 80% of the time. In other words, 80% of the time each person is assigned to a shift or training they are paired with the same partner.

6.2.2 Functional Requirements (Mandatory Training)

6.2.2.1 The model shall schedule monthly and annual required training. Monthly mandatory training consists of TR, T1, T3 and T4. Annual evaluations required for personnel consist ofa four-hour Annual Evaluation in the simulator. (see Table 1)

6.2.2.2 The model shall schedule one or two instructors for each monthly training event.

6.2.2.3 The model shall schedule three evaluators for each annual evaluation.

6.2.3 Functional Requirements (General)

6.2.3.1 The model shall be able to rebuild the schedule anytime during the calendar month due to an unforeseen absence of an individual.

6.2.3.2 While rebuilding the schedule, the model shall minimize rescheduling of future events due to an unforeseen absence of an individual.

6.2.4 Technical Requirements

6.2.4.1 The model shall be able to run on a server.

6.2.4.2 The model that runs on a server shall be accessible from a client workstation.

6.2.4.3 The model shall provide an output in a format that can be input to TimePiece.

6.2.4.4 The model shall be able to import and export data in XML, CSV, and XLSX formats.

6.2.5 Project Requirements

6.2.5.1 Team AFOCS shall develop an optimization algorithm/model.

6.2.5.2 Team AFOCS shall compare performance of different algorithms/approaches.

6.2.5.3 Team AFOCS shall develop a Requirements Document.

7. The Model

This section details the model's main structure in terms of the decisions made, the constraints on the decisions, and the process for performing the crew assignments. As described in Section 5.3 Algorithm Development, the model executes in four phases to create the original schedule, and has two methods to address daily changes to personnel availability. Each phase consists of running a mathematical program to perform a series of assignments, and contains several parameter assignments to backup certain variables to be used on other mathematical program executions.

The following subsections describe each phase and method individually with the final section describing the model execution process. To understand the phases and methods of the model, the first section describes the main data elements contained in the model. It should be noted that there are additional data elements, variables, and constraints not contained in this report that are necessary for the model to run appropriately. The entire model structure is contained in the Model Documentation/Code appendix (see 12.2 Model Documentation/Code).

In developing the optimization algorithm, the model, the AFOCS team has the following assumptions:

- The optimization algorithm will have, at minimum, the following data for input:
    - Calendar month/year to be scheduled
    - Pre-scheduled days for training classes
    - Number of seats available in each classroom
    - Classroom availability for trainees
    - Number of INSTs required for the simulation training event TR

- Number of slots per day available for the simulation training events TR and AE (Simulator availability)
- Number of personnel, their squadron, job title, functional role, and "crewmate"
- Unavailable days for personnel due to Leave, Duty Not Including Alert (DNIA), Temporary Duty (TDY), etc. Personnel can be available for training on days they are on DNIA but cannot be assigned to a shift on that day.

7.1 Main Data Elements

The main data elements necessary to understand the assignment phases and dynamic methods of the model are described below:

- Personnel (p, p2) = all personnel to be scheduled during the month
  - Crew members (cw) = subset of personnel, contains all personnel designated as crew members for their job title
  - Instructors (ins) = subset of personnel, contains all personnel designated as instructors for their job title
  - Evaluators (eval) = subset of personnel, contains all personnel designated as EVALs for their job title
  - Flight Commanders (flt cdr) = subset of personnel, contains all personnel designated as FLT CDRs for their job title
- Site (site) = all sites to be scheduling during the month
  - SCP sites (scp) = subset of sites, contains all SCP sites
- Squadron (sq) = the three squadrons under the Group's supervision
- Events (e) = all events to be scheduled during the month, contains {Alert, O-day, B-day, TR, T1, T3, T4, T3/T4, AE, backup, leave}
  - Classroom Events (class) = subset of events, corresponds to T1, T3, T4, T3/T4
  - Simulation Events (simEvent) = subset of events, corresponds to TRs and AEs
  - Training Events (train) = subset of events, corresponds to class and TR events
- Jobs (job) = job titles, corresponds to CW, INST, EVAL, Flight CDR
- Position (pos) = position title, corresponds to CDR and DEP
- Slots (sl) = slots for simulation training events, contains sl1, sl2, sl3, sl4, sl5
- Simulators (sim) = simulators used for simulation training events, contains sim1, sim2
- Days (d, d2) = all days 1-31
- Month (m) = all months 1-12

Personnel are mapped to squadrons, jobs, and positions by binary parameters where a 1 indicates that the person has a relationship to that other data element. For example, PersToPos(p,pos) maps a person to a position with PersToPos(p,pos) set to 1 if person p has position pos. These binary parameters are necessary when assigning personnel to ensure that personnel (1) are assigned to shifts and training to meet the requirement and (2) are not scheduled for an assignment in which they do not meet the qualifications to perform that assignment.

Sites are mapped to squadrons through a binary parameter similar to the assignment of personnel to other data elements. Each squadron has 5 sites, with 1 site being the SCP site.

7.2 Assign Alerts

The Assign Alerts phase performs the assignment of personnel to shifts at each of the 15 sites. Each site requires one CDR and one DEP each day, with the possibility of having a CDR fill the DEP position. Personnel must have at least 3 days between shifts. For example, if a person has shift on day 2, then the next day they can be on shift is day 5. The day after a person is on shift is scheduled as their "O-day", or off-day. The day after their "O-day" is scheduled as their "B-day", which is a placeholder to ensure no alerts are assigned on that day. The "B-day" assignment is removed after all shift assignments are completed. The following describes the primary variables and constraints used during the Assign Alerts phase:

Variables

- AssignPersToSite(p, d, site) = 1 if person p is assigned to that site on day d, 0 otherwise
- PersonSchedule(p, d, e) = 1 if person p is assigned to event e on day d, 0 otherwise
- ObjectiveFunctionAlert = objective function for this phase, the objective is to maximize crew integrity (the % of times personnel are assigned with their crewmate) and spread out the alert assignments among the personnel within the same job category

Constraints

- Personnel can only be assigned to a site within their squadron
- Each site must have one CDR and one DEP assigned each day, with the exception that another CDR can fill the DEP slot
- Only personnel with SCP certification can be assigned to a SCP site
- The day after a person is on shift is their O-day
- The day after a person's O-day is their B-day
- Personnel can only be assigned to one site per day
- Personnel can only be assigned to one event per day
- Personnel can only be assigned for a shift on days they are available to be on alert

A parameter, named PersonAlertSchedule(person,day,event), is set for each person's shift and O-day schedule after this phase executes. The PersonAlertSchedule parameter is used during the Assign Training phase to ensure the AssignTrainingSchedule mathematical program does not override the AssignAlertSchedule mathematical program assignments. Also at the end of the AssignAlertSchedule execution, the B-day assignments are removed because a person can perform training on that day whereas they cannot pull a shift.

7.3 Assign Training

The Assign Training phase performs the assignment of personnel, INSTs, and EVALs to classroom and simulation training in fulfillment of monthly training and annual evaluation requirements. Each person is required to complete the T1, T3, T4, and TR training events each month. The T3 and T4 training classes are either held separately or together on the same day, thus the reason for the T3/T4 training event. If a person attends the T3/T4 class, then they

receive credit for both T3 and T4 training classes. The following describes the primary variables and constraints used during the Assign Training phase:

Variables

- AssignPersonToTrainingClass(p,d,class) = 1 if person p assigned to classroom training class on day d, 0 otherwise
- AssignPersonToSimTraining(p,d,simEvent) = 1 if person p assigned to that simulation training simEvent on day d, 0 otherwise
- AssignInstuctorsToTrainClass(ins,d,class) = 1 if instructor ins assigned to teach classroom training class on day d, 0 otherwise
- AssignInstructorsToTrainTRs(ins,d,simEvent) = 1if instructor ins assigned to teach simEvent "TR" on day d, 0 otherwise
- AssignEvaluatorsToEvals(evals,d,simEvent) = 1 if evaluator evals assigned to simEvent "eval" on day d, 0 otherwise
- ClassRequirement(p,class) = 1 if person p has fulfilled the training requirement to take classroom training class, 0 otherwise
- ObjectiveFunctionTraining = objective function for this phase, , the objective is to (1) maximize crew integrity (the % of times personnel are assigned with their crewmate), (2) spread out classroom and TR teaching assignments across INSTs, and (3) spread out annual evaluation assignments across EVALs

Constraints

- Assignment schedules must adhere to the shift and O-day assignments performed during the Assign Alerts phase (no alerts or O-days can be overridden)
- All personnel must meet monthly training class requirements
- Personnel can only be assigned to training events on days training events are held (weekdays for classrooms, weekdays and Saturdays for TRs and AEs)
- Number of personnel assigned to training classes cannot exceed classroom size
- INSTs can only be assigned to TR simulation events
- EVALs can only be assigned to AE simulation events
- INSTs can either attend or teach a particular training event per day
- EVALs can either attend or teach annual evaluation events per day
- INSTs can teach a maximum of 2 TRs per day
- The number of people assigned to TRs and AEs per day are limited by the number of slots available and the number of simulators available
- TRs require a specific number of INSTs for each TR slot and simulator combination (the specific number is a user input, default is 2)
- Annual evaluations require 3 EVALs for each AE slot and simulator combination
- At most 1 CDR and 1 DEP can be assigned to a simulation event slot and simulator combination
- Limit the number of personnel with alert the next day to the number of personnel that can be assigned in the first 3 slots (ensures personnel with alert the next day are not in training within 12 hours of going on alert)

- Limit the number of INSTs and EVALs with alert the next day to the number of INSTs and EVALs that can be assigned in the first 3 slots to teach simulation training events (ensures INSTs and EVALs with alert the next day are not performing training within 12 hours of going on alert)
- Personnel can only be assigned to training on days they are available for training
- Personnel can only be assigned to one event per day

Two parameters, named PersonClassSchedule(person,day,class) and PersonSimEventSchedule(person,day,simEvent), are set for each person's training schedule after this phase executes. These parameters are used during the Assign Backup Crew and Assign Slot phases to ensure the AssignBackupCrewSchedule and AssignSlotSchedule mathematical programs do not override the AssignTrainingSchedule mathematical program assignments.

7.4 Assign Backup Crew

The Assign Backup Crew phase performs the assignment of personnel to the backup crew. The backup crew is used when a person who is assigned for a shift on a particular day becomes unavailable to pull that shift. The use of the backup crew is further described later during the Heuristic method approach to responding to unforeseen personnel unavailability. The following describes the primary variables and constraints used during the Assign Backup phase:

Variables

- PersAssignedAsBackupCrew(p,d) = 1 if person p is assigned to the backup crew on day d

Constraints

- Assignment schedules must adhere to the Assign Alert and Assign Training phase assignments
- Must have 1 DEP and 1 CDR on the backup crew
- Personnel cannot be assigned to the backup crew the day after they have an O-day
- Personnel cannot be assigned to the backup crew if they have a shift within the next 2 days, so if they were pulled to active duty on the day they are on backup it would not conflict with the requirement to have 3 days between alerts
- Personnel cannot be assigned to the backup crew if the total number of shifts they are assigned to plus days on backup they are assigned to exceeds the maximum number of shifts their job title can pull
    - For example, if an INST is assigned to 2 shifts during the month, then they cannot be assigned to backup because if they were on backup and required to pull a shift, then they would exceed the maximum number of shifts allowed for INSTs in a given month
- Personnel cannot be assigned to the backup crew if they are unavailable for alert or on leave
- Personnel can only be assigned to one event per day

The Assign Backup Crew phase uses the objective function from the Assign Training phase.

A parameter, named PersBackupSchedule(person,day), is set for each person's backup schedule after this phase executes. The PersBackupSchedule parameter is used during the Assign Slots phase to ensure the AssignSlotSchedule mathematical program does not override the AssignBackupCrewSchedule mathematical program assignments.

7.5 Assign Slots

The Assign Slots phase performs the assignment of personnel, INSTs, and EVALs to simulator and slot combinations to conduct the TR and AE simulation events. There can either be four or five available slots per day, each with one or two simulators available in each slot. The following describes the primary variables and constraints used during the Assign Slots phase:

Variables

- AssignPersToSlot(sl,sim,d,p) = 1 if person p is assigned to slot sl and simulator sim on day d, 0 otherwise
- AssignInstructorsToSlot(sl,sim,d,ins) = 1 if instructor ins is assigned to slot sl and simulator sim on day d, 0 otherwise
- AssignEvaluatorsToSlot(sl,sim,d,evals) = 1 if evaluator evals is assigned to slot sl and simulator sim on day d, 0 otherwise
- SlotUsedForSimEvent(sl,sim,d,simEvent) = 1 if the slot sl and simulator sim combination is being used for simulation training simEvent on day d, 0 otherwise

Constraints

- Assignment schedules must adhere to the Assign Alert, Assign Training, and Assign Backup Crew phase assignments
- A person must be assigned to a slot and simulator combination if they are assigned to a simulation event on that day
- The number of INSTs or EVALs assigned to a slot and simulator combination must meet the requirement for the simulation event being performed on that day
- A slot and simulator combination can only be used for one simulation training event each day
- INSTs can only be assigned to a slot and simulator combination where the combination is being used for TRs
- EVALs can only be assigned to a slot and simulator combination where the combination is being used for annual evaluations
- INSTs and EVALs can only be assigned to one simulator in each slot
- INSTs can only be assigned to at most two slots, which must be sequential
- At most 1 DEP and 1 CDR can be assigned to a slot and simulator combination
- INSTs and EVALs can either attend or teach in a particular slot and simulator combination

The Assign Slots phase uses the objective function from the Assign Training phase.

7.6 Heuristic Approach to Unforeseen Personnel Unavailability

This subsection describes the heuristic approach to address the situation where a person is scheduled for a shift on a particular day and can no longer go on alert. The heuristic approach performs the following steps:

1. Change person availability for the person now unavailable on the "problem day"
2. Pull the backup crew member with the same position as the person now unavailable
3. Assign the backup crew member to the site the person now unavailable was assigned to
4. Change the alert and O-day schedules for the backup crew member and the person now unavailable to match the new assignments
5. Run the AssignBackupCrewSchedule procedure to fill in the backup crew slot that is now vacant due to the backup crew member being pulled on shift

7.7 Dynamic Approach to Unforeseen Personnel Unavailability

This subsection describes the dynamic approach to address the situation where more than 2 people are scheduled for shifts on a particular day and can no longer go on alert. The dynamic approach is different than the heuristic approach in that the backup crew cannot fill the demand created from multiple personnel becoming unavailable on a single day. The dynamic approach must change multiple schedules and still ensure that all the scheduling requirements and original constraints are satisfied. This approach has variables that calculate the difference between the original schedule and the new schedule being formed. The goal is to minimize the difference between the original schedule and the new schedule while still adhering to all constraints from the original schedule assignments. The dynamic approach is summarized as follows:

1. SetCompletedSchedules – grabs the original schedule from day one of the month up to the day before the "problem day". This schedule will be used as input to the dynamic mathematical procedures to ensure that the portion of the schedule that has been completed is not altered
2. SetNewPersonAvailability – after user inputs the personnel who are now unavailable, this sets the personnel availability parameters used during scheduling assignments
3. EmptyAllOriginalVariables – empties all the original variables with the original schedules saved in backup parameters
4. PerformDynamicAlertPortion – performs the Assign Alert phase with a modified objective function to minimize the difference between the original schedule and the new schedule
5. PerformDynamicTrainingPortion – performs the Assign Training and Assign Backup Crew phases with a modified objective function to minimize the difference between the original schedule and the new schedule
6. PerformDynamicSlotPortion – performs the Assign Slot phase with a modified objective function to minimize the difference between the original schedule and the new schedule

The above procedures described in steps 1-7 are contained in the procedure DynamicMainExecution.

7.8 Model Execution

The phases to create the original schedule executes in the following process:

1. Main Initialization – reads the input data, sets the slot and simulator day availability based upon the input month and input year, sets the shift requirement for each site
2. Main Execution – main procedure to execute, consists of the following procedures:
    a. AssignAlertSchedule – performs the Assign Alert phase
    b. AssignTrainingSchedule – performs the Assign Training phase
    c. AssignBackupCrewSchedule – performs the Assign Backup Crew phase
    d. AssignSlotSchedule – performs the Assign Slots phase
    e. SetOriginalSchedule – backups all variables from the assignment phases, used during the dynamic portion
3. If the personnel availability changes during the month, then perform the following:
    a. If 1-2 people become unavailable on a day, run AssignBackupCrewToFillDemand
    b. If more than 2 people become unavailable, run DynamicMainExecution

8. Testing and Validation

This section describes the testing and validation approach to ensure the model was built correctly.

8.1 Testing

Model testing encompasses two different activities: (1) ensuring the model met the requirements stated in the SRD, and (2) comparing solver runtime against best solution obtained. These two testing activities test the model logic and the model performance. Testing the model performance is important since the original need for the model was to improve the efficiency of the current scheduling process; therefore the model must perform faster than the current process to have improved upon the current process. Testing the model logic is equally important as the model must be useful to the user. If the model performs quickly but does not output the correct schedule or does not meet the stated requirements, then the model is not a useful tool for scheduling personnel.

The first portion of the model testing entails running the model and comparing the model outputs relative to SRD requirements. If the model meets the requirement, the test is considered passed. If not, then the test is considered a failure. Any failed tests require attention to remedy the portion of the model that is not executing correctly. Any model modifications must be retested to ensure the requirement is fulfilled. The model is complete when it meets all stated requirements. The model passed all tests, meeting all requirements in the SRD, and thus is determined to be complete according to the requirements set forth before model development.

The second portion of the model testing entails running the model for a specified amount of time and comparing how close the best solution found is to the upper bound. Since the model is a mixed-integer program (MIP), the solver must perform various techniques such as branch-and-bound and cut-generation to find a feasible integer solution. The solver will originally perform a

linear program (LP) relaxation, which means it will treat all variables as continuous variables. The solution to the LP-relaxation problem becomes the upper bound for the integer solution. The solver will then perform various methods to find the best integer solution and attempt to reduce the gap between the latest integer solution found and the LP upper bound. Finding the best integer solution can take a long time, especially with as many binary variables contained in this model. Therefore the solver runtime, or the amount of time the solver is allowed to run and find a solution, is limited.

In testing the performance of the model, it was found that the Assign Alert phase took the longest amount of time to solve. The other phases executed in under 1minute. Therefore the Assign Alert phase mathematical program was the only mathematical program tested at various solver runtimes. Table 2 below details the solver runtimes and gap percentage between the LP upper bound and the optimal integer solution for the phases other than Assign Alert. The gap percentage is the percentage difference between the LP upper bound and the best integer solution found. Table 3 and Figure 2 below detail the performance test results from multiple runs of the Assign Alert phase mathematical program at various solver runtimes. The figure shows the tradeoff between solver runtime (>5 min) and gap percentage.

As shown in the table and the figure, running the model for 4 hours (240 minutes) does not improve upon the solution found after 3 hours (180 minutes), thus the model should not run past 3 hours. The remaining phases take less than 1 minute to run, thus the total model runtime is approximately 3 hours.

The model performs better than the current scheduling process, which takes approximately 1.5 weeks to complete. The model reduces the scheduling process time down to approximately 3 hours and finds a solution within 2% of optimality.

Table 2 Performance Metrics for Mathematical Programs

Mathematical Program	Best LP Bound	Best Solution	Gap (%)	Solving Time (sec)	Peak Memory (Mb)
AssignTrainingSchedule	1807.2	1806.3	0.1	43.5	267.8
AssignBackupCrewSchedule	1806.3	1806.3	0.0	0.9	40.3
AssignSlotSchedule	1554.3	1554.3	0.0	4.4	169.3
AssignBackupCrewToFillDemand	1554.3	1554.3	0.0	1.0	-

Table 3 Performance Metric Comparison for Runtimes for Assign Alerts Phase Mathematical Program

Runtime (min)	Best LP Bound	Best Solution	Gap (%)	Solving Time (sec)	Peak Memory (Mb)
5	909.1	360.1	152.6	301.7	251.4
10	909.1	788.1	15.4	602.1	340.0
20	909.1	797.2	14.0	1201.5	464.8
30	909.0	797.2	14.0	1802.5	485.9
40	908.9	824.4	10.2	2402.1	512.1
50	908.9	824.4	10.2	3001.9	507.2
60	908.9	824.4	10.2	3602.5	512.1

Runtime (min)	Best LP Bound	Best Solution	Gap (%)	Solving Time (sec)	Peak Memory (Mb)
120	908.9	880.8	3.19	7237.3	1823.1
180	908.9	891.3	1.97	10822.5	1892.2
240	908.9	891.3	1.97	14415.1	-

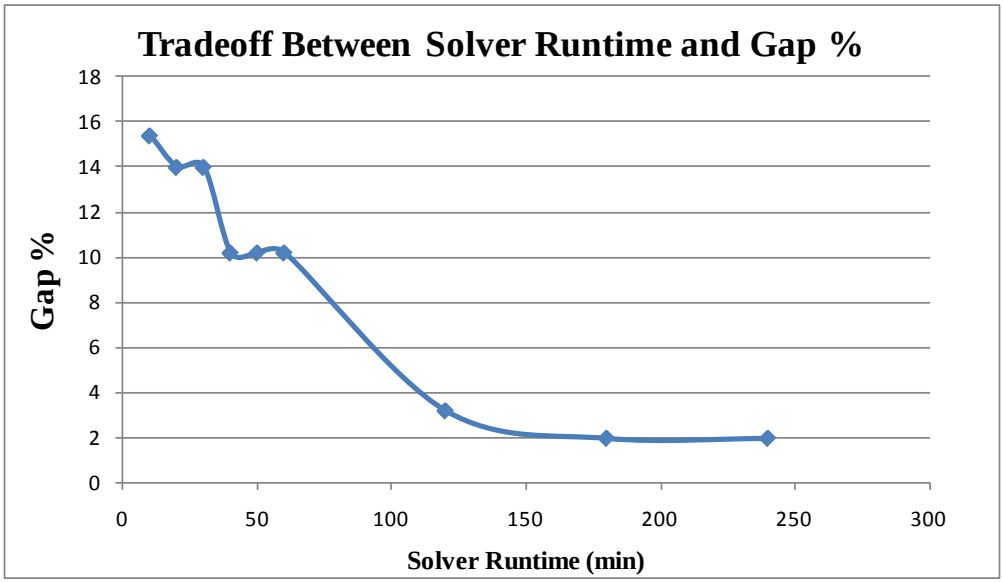


Figure 2 Tradeoff Between Solver Runtime and Gap % for Assign Alerts Phase Mathematical Program

8.2 Validation

To validate our model, the Requirements Verification Matrix technique was used. See appendix 12.3 Requirements Verification Matrix for the full matrix.

9. Recommendations

A multi-location scheduling effort that takes into account shifts, allocated time off and training for CWs, INSTs, EVALs and FLT CDRs takes 1.5 weeks to manually create the schedule. This process needs to be automated and optimized to reduce the cycles used to schedule the operations centers. This can be accomplished today through an optimized model for automatic scheduling for operations centers to improve efficiency and performance of the existing scheduling process.

AFOCS developed a model using AIMMS that uses an optimized algorithm to create the schedule for the operations centers while remaining dynamic in nature. The exploration of algorithms was covered in the literature review. AFOCS recommends the incorporation of the model developed using AIMMS be incorporated into the scheduling process for the operations centers.

10. Future Work

The project's scope was limited to developing, testing and analyzing an optimization algorithm to improve efficiency and performance of the existing scheduling process. Below are possible work efforts that could be implemented in the future:

- To broaden the span, multiple month schedules can be considered. In the current model, the schedule is created for one month. Realistically, the schedule for two months out is likely to be impacted by a schedule change in the current month's schedule.

- Implementation Research/Design. This future work could continue to investigate how to take the outputs of AIMMS to feed into TimePiece for the model to be incorporated into the operations centers scheduling effort.

- Test Procedures. The current model will need a test plan that includes test procedures developed to test with the client's system.

- Additional Constraints. As the client begins using the system, more requirements may be developed that will need to be incorporated into the model.

- Additional Modeling. This future work would be to expand the model based off of client feedback to broaden its applicability to more than just the op centers but to other clients of Kepler Research.

- Architecture Diagram. This future work would be to develop an architecture diagram using a data model to show the different pieces of data and how the pieces flow together. This will help ease the tracking of the outputs from the model and where the outputs originate.

11. References

Collins, B., & Prebble, M. (2011). *Scheduling Requirements for GMU Kepler Senior Project.* Kepler Research.

Paragon Decision Technology. (n.d.). *AIMMIS Optimization Software for Mathematical Programming*. Retrieved from http://www.aimms.com

[This page intentionally left blank.]

12. Appendices

12.1 Project Management

Work Breakdown Structure (WBS)

The AFOCS project was broken into 6 different phases; shown in Figure 3.

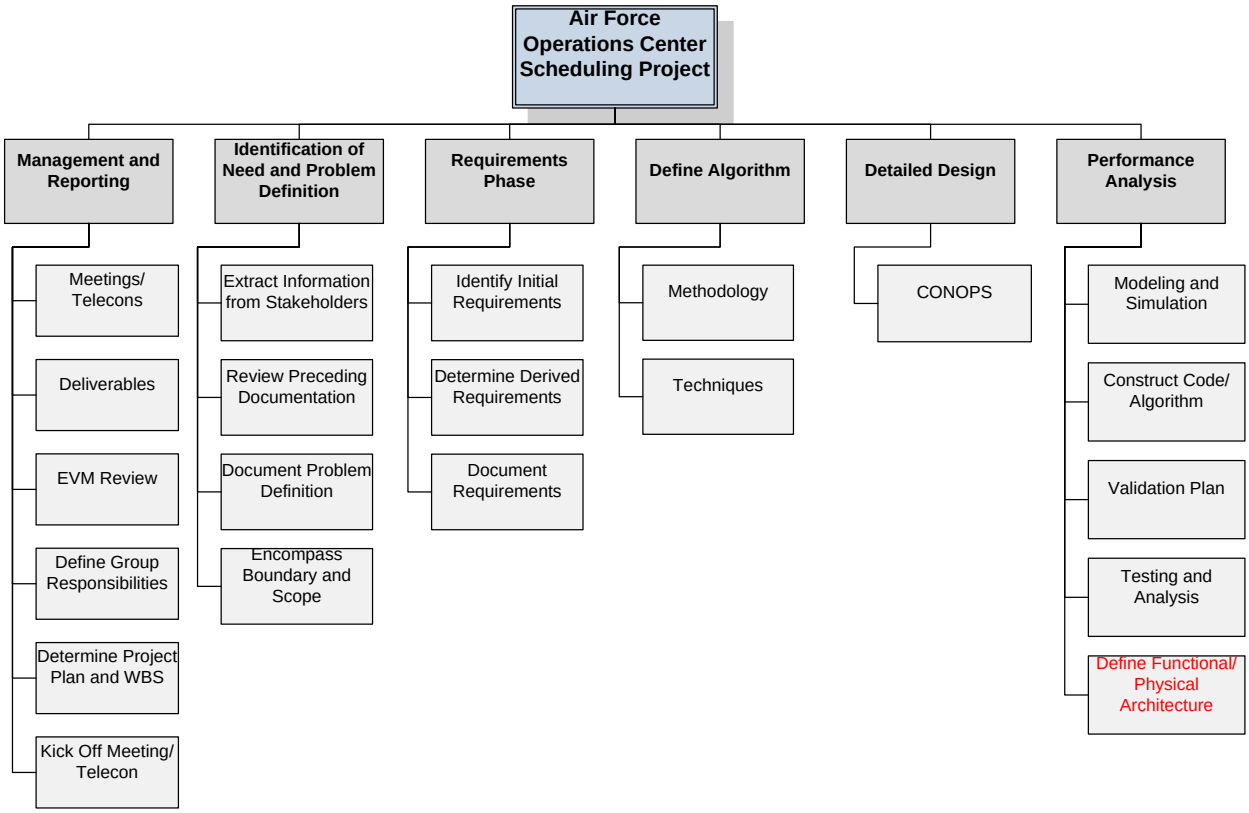


Figure 3: WBS of the AFOCS Project

GANTT Chart

The AFOCS team created a work breakdown structure using Microsoft Project to track the completion of the project through the spring 2011. The tasks for the project are shown in Figure 4 while the associated Gantt chart is shown in Figure 5.

AFOCS_Project Plan

ID	Task Name	Duration	Start	Finish	Predecess	Resource Names
1	**Air Force Operations Center Scheduling Project**	**568 hrs?**	**Wed 2/2/11**	**Tue 5/10/11**		
2	**Management and Reporting**	**568 hrs**	**Wed 2/2/11**	**Tue 5/10/11**		
3	Kick-Off Telecon	2 hrs	Wed 2/2/11	Wed 2/2/11		Amar Zabarah,Ashley
4	Determine Project Plan and Work Breakdown Structure	4 hrs	Thu 2/3/11	Thu 2/3/11		Amar Zabarah
5	Define Group Responsibilities	4 hrs	Mon 2/7/11	Mon 2/7/11		Ashley Rock,Rebecc
6	Kick Off Meeting	2 hrs	Mon 2/7/11	Mon 2/7/11		Rebecca McCrabb,Sc
7	EVM Review	16 hrs	Wed 2/2/11	Tue 2/15/11		Amar Zabarah
8	**Meetings/Telecon (Biweekly)**	**496 hrs**	**Wed 2/2/11**	**Thu 4/28/11**	35	
16	**Deliverables**	**560 hrs**	**Thu 2/3/11**	**Tue 5/10/11**		
17	Proposal	32 hrs	Fri 2/11/11	Thu 2/17/11	37	Scott Genberg[20%]
18	Interim Progress Report I	72 hrs	Mon 2/21/11	Thu 3/3/11		Amar Zabarah[10%],A
19	Interim Prohress Report II	72 hrs	Tue 4/19/11	Fri 4/29/11		Amar Zabarah[10%],A
20	Optimization algorithm(s)	72 hrs	Thu 4/28/11	Tue 5/10/11		Rebecca McCrabb[25
21	Website	72 hrs	Tue 4/26/11	Fri 5/6/11		Amar Zabarah[25%]
22	**Presentation that communicates results**	**544 hrs**	**Thu 2/3/11**	**Fri 5/6/11**		
23	**Status Report (Biweekly)**	**490 hrs**	**Thu 2/3/11**	**Thu 4/28/11**		
31	Dry Run	4 hrs	Thu 4/28/11	Thu 4/28/11		
32	Final Presenation	8 hrs	Fri 5/6/11	Fri 5/6/11	31	
33	Final Techincal Report	72 hrs	Tue 4/26/11	Fri 5/6/11		
34	**Identification of Need and Problem Definition**	**69 hrs**	**Mon 2/7/11**	**Thu 2/17/11**		
35	Extract Information from Stakeholders	2 hrs	Mon 2/7/11	Mon 2/7/11		Rebecca McCrabb,Sc
36	Review Preceding Documentation	4 hrs	Fri 2/11/11	Wed 2/16/11	35	Ashley Rock,Amar Za
37	Document Problem Definition	48 hrs	Tue 2/8/11	Wed 2/16/11	35	Scott Genberg[20%]
38	**Encompass Boundary and Scope**	**67 hrs**	**Tue 2/8/11**	**Thu 2/17/11**		
39	Define Objectives	4 hrs	Tue 2/8/11	Tue 2/15/11	6,35	Rebecca McCrabb,Sc
40	Define Assumptions	2 hrs	Thu 2/17/11	Thu 2/17/11	35,36,37	Amar Zabarah,Ashley
41	Define Constraints	4 hrs	Mon 2/14/11	Thu 2/17/11	35,36,37	Amar Zabarah,Ashley
42	**Requirements Phase**	**27.9 hrs**	**Mon 2/14/11**	**Thu 2/17/11**		
43	Identify Initial Requirements (originating)	16 hrs	Mon 2/14/11	Tue 2/15/11	35	Scott Genberg
44	Determine Derived Requirements	12 hrs	Mon 2/14/11	Tue 2/15/11		Amar Zabarah,Ashley
45	Document Preliminary Requirements	12 hrs	Tue 2/15/11	Thu 2/17/11	43,44	Scott Genberg
46	**Define Algorithm**	**76 hrs**	**Mon 2/21/11**	**Fri 3/4/11**		
47	Methodology	48 hrs	Mon 2/21/11	Mon 2/28/11	40,41,45	Amar Zabarah[10%],F
48	Techniques	28 hrs	Tue 3/1/11	Fri 3/4/11	47	Amar Zabarah,Rebec
49	**Detailed Design**	**76 hrs**	**Mon 2/28/11**	**Fri 3/11/11**		
50	Concept of Operations (CONOPS)	52 hrs	Thu 3/3/11	Fri 3/11/11	40,41,45	Amar Zabarah[20%],A
51	Determine and Document Final Mission Requirements	36 hrs	Mon 2/28/11	Fri 3/4/11		Scott Genberg[10%]
52	**Performance Analysis**	**248 hrs?**	**Thu 3/10/11**	**Thu 4/21/11**		
53	Modeling and Simulation	64 hrs	Thu 3/10/11	Mon 3/21/11	40,47	Rebecca McCrabb[20
54	Construct Code/Algorithm	24 hrs	Mon 3/14/11	Wed 3/16/11	40,47	Rebecca McCrabb
55	Validation Plan	32 hrs	Mon 3/21/11	Thu 3/24/11		Ashley Rock,Amar Za
56	**Define Functional/Physical Architecture**	**144 hrs**	**Tue 3/22/11**	**Thu 4/14/11**		
57	Define Operational Architecture	72 hrs	Mon 4/4/11	Thu 4/14/11		Ashley Rock[20%],Sc
58	Requirements Traceability	44 hrs	Tue 3/22/11	Tue 3/29/11	53,54	Ashley Rock[50%],Sc
59	**Analysis**	**184 hrs?**	**Tue 3/22/11**	**Thu 4/21/11**		
60	Evaluate Algorithm	8 hrs?	Tue 3/22/11	Tue 3/22/11	53,54	Amar Zabarah,Rebec
61	Testing	56 hrs	Thu 3/24/11	Fri 4/1/11	53	Amar Zabarah,Rebec
62	Decision Analysis	64 hrs	Tue 4/12/11	Thu 4/21/11	53,54	Amar Zabarah
63	Cost Analysis (Maybe)	10 hrs	Mon 4/11/11	Tue 4/12/11		

Figure 4: AFOCS project schedule

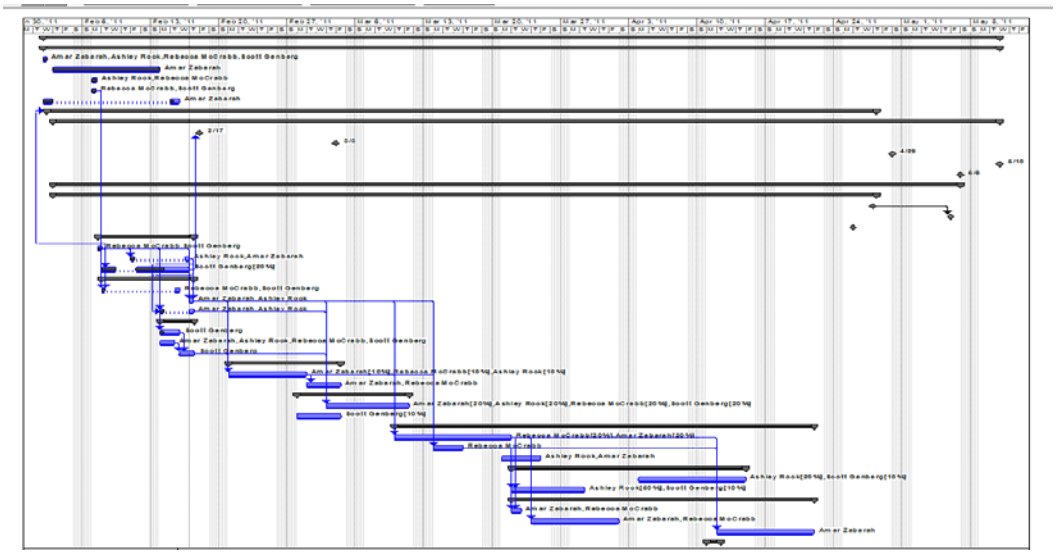


Figure 5: AFOCS Gant Chart

Earned Value Management (EVM)

Using Project Management, the AFOCS team assigned resources and a budget for individual tasks. Also, the AFOCS team assigned overall budget of approximately $ 30,000. The EVM chart to keep track of the tasks and dates was originally produced on 10 February 2011, as seen in Figure 6. Earned Value Management was calculated weekly and reported graphically.

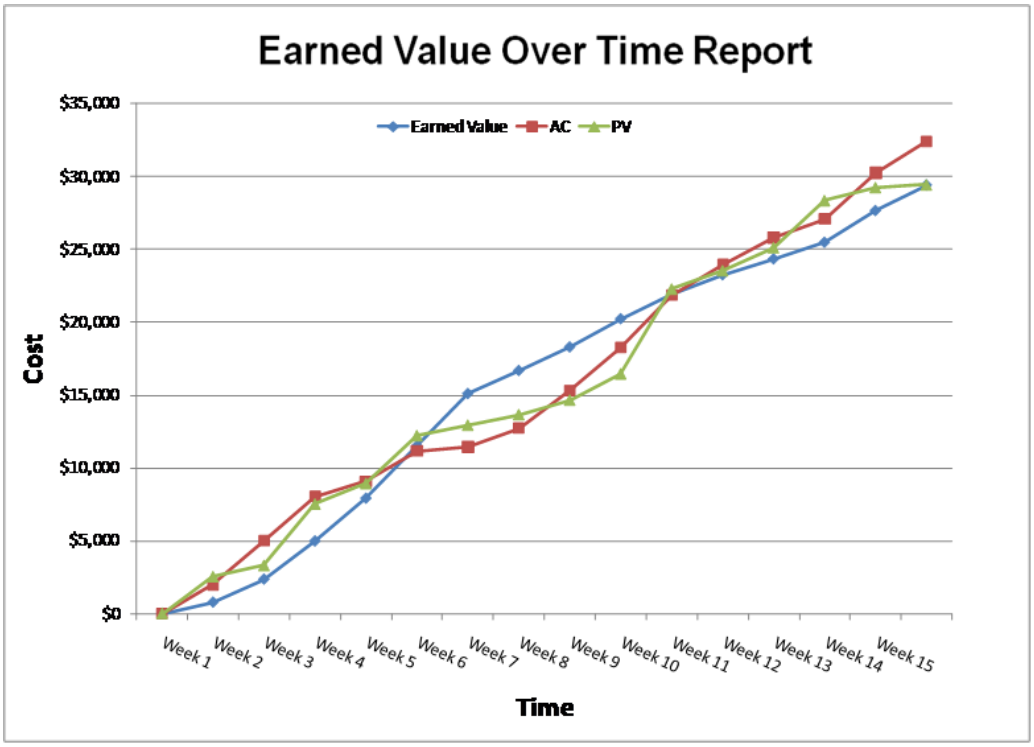


Figure 6: AFOCS EVM as of February 10th 2011

Roles and Responsibilities

Name	Major	Primary Task	Secondary Task
Scott Genberg	Systems Engineering	Requirements	Project Management
Rebecca McCrabb	Operations Research	Model Development	Project Management
Ashley Rock	Systems Engineering	Literature Review	Project Management
Amar Zabarah	Operations Research	Website and Documentations	Project Management

Time Sheets

Below are the team's time sheets, one as a whole and the rest are per team member. (The column highlighted in yellow is spring break)

The Team:

Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		1/24/2011	1/31/2011	2/7/2011	2/14/2011	2/21/2011	2/28/2011	3/7/2011	3/14/2011	3/21/2011	3/28/2011	4/4/2011	4/11/2011	4/18/2011	4/25/2011	5/2/2011	
WBS	Name:	AFOCS															Total
1.1	Management	11	7	7	6	2	1	2	1	2	3	2	2	2	2	0	50
1.2	Literature Review	0	0	3	8	9	10	7	3	8	11	3	2	4	3	0	71
1.3	Requirements	2	4	3	4	4	8	4	2	4	3	4	1	3	1	0	47
1.4	Design	7	6	5	3	5	5	0	0	0	0	0	0	0	0	0	31
1.5	Modeling and Simulation	0	0	5	7	10	10	10	10	10	10	10	10	10	5	0	107
1.6	Test	0	0	0	0	0	0	0	0	0	0	1	1	5	6	0	13
1.7	Analysis	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
1.8	Presentations	2	5	3	4	3	4	5	0	0	6	1	0	4	3	0	40
1.9	Documentation	4	5	8	7	2.5	4	3	9	10	2	17	22	18	40	0	151.5
	TOTAL	26	27	34	39	36	42	31	25	34	36	38	38	46	60	0	511.5

Scott Genberg

Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		1/24/2011	1/31/2011	2/7/2011	2/14/2011	2/21/2011	2/28/2011	3/7/2011	3/14/2011	3/21/2011	3/28/2011	4/4/2011	4/11/2011	4/18/2011	4/25/2011	5/2/2011	
WBS	Name:	Scott Genberg															Total
1.1	Management				1	1	1	1	1	2	1	1	1	1	1		12
1.2	Literature Review				1					1	1	1	1	1			6
1.3	Requirements			3	2	2	4	4	2	4	3	2	1	2			29
1.4	Design																0
1.5	Modeling and Simulation																0
1.6	Test											1	1	1			3
1.7	Analysis										1						1
1.8	Presentations							1			2			2	1		6
1.9	Documentation			3	2	2	4	3	1	2	2	4	3	5	10		41
	TOTAL	0	0	6	6	5	9	9	4	9	10	9	7	12	12	0	98

Rebecca McCrabb

Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		1/24/2011	1/31/2011	2/7/2011	2/14/2011	2/21/2011	2/28/2011	3/7/2011	3/14/2011	3/21/2011	3/28/2011	4/4/2011	4/11/2011	4/18/2011	4/25/2011	5/2/2011	
WBS	Name:	Rebecca McCrabb															Total
1.1	Management	3	1														4
1.2	Literature Review																0
1.3	Requirements	2	4														6
1.4	Design	5	5	5	3												18
1.5	Modeling and Simulation			5	7	10	10	10	10	10	10	10	10	10	1		103
1.6	Test													4	6		10
1.7	Analysis																0
1.8	Presentations																0
1.9	Documentation														11		11
	TOTAL	10	10	10	10	10	10	10	10	10	10	10	10	14	18	0	152

Ashley Rock

Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		1/24/2011	1/31/2011	2/7/2011	2/14/2011	2/21/2011	2/28/2011	3/7/2011	3/14/2011	3/21/2011	3/28/2011	4/4/2011	4/11/2011	4/18/2011	4/25/2011	5/2/2011	
WBS	Name:	Ashley Rock															Total
1.1	Management	3	1			1		1				1	1	1	1		10
1.2	Literature Review			3	5	7	8	7	3	7	10	2	1	3	3		59
1.3	Requirements																0
1.4	Design	2	1														3
1.5	Modeling and Simulation																0
1.6	Test																0
1.7	Analysis																0
1.8	Presentations	2	4	2	3	3	4	4				1			2		25
1.9	Documentation	4	5	5	3							5	8	5	5		40
	TOTAL	11	11	10	11	11	12	12	3	7	10	9	10	9	11	0	137

Amar Zabarah

Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		1/24/2011	1/31/2011	2/7/2011	2/14/2011	2/21/2011	2/28/2011	3/7/2011	3/14/2011	3/21/2011	3/28/2011	4/4/2011	4/11/2011	4/18/2011	4/25/2011	5/2/2011	
WBS	Name:	Amar Zabarah															Total
1.1	Management	5	5	7	5						2						24
1.2	Literature Review				2	2	2										6
1.3	Requirements				2	2	4					2		1	1		12
1.4	Design					5	5										10
1.5	Modeling and Simulation														4		4
1.6	Test																0
1.7	Analysis																0
1.8	Presentations		1	1	1						4			2			9
1.9	Documentation				2	0.5			8	8		8	11	8	14		59.5
	TOTAL	5	6	8	12	9.5	11	0	8	8	6	10	11	11	19	0	124.5

12.2 Model Documentation/Code

Design and development of the optimization algorithm, the model, is conducted through an iterative approach. The model is developed using Linear Programming and Integer Programming techniques. The software tool that is used is AIMMS v3.11 (www.aimms.com).

The AFOCS Model was broken down into different sections:

1- Model Data
2- Model Variables
3- Model Constraints
4- Initialization Procedures
5- Mathematical Program Procedures
6- Outputs
7- Sanity Checks

Each section will be explained in details (Initialization Procedures and Mathematical Program Procedures will be covered in one section, see 12.2.5 Model Execution).

The objective is to create an optimized staffing schedule that satisfies all the requirements that were collected from the customer (see 6 Requirements).

12.2.1 Model Data

Assumptions

The following are assumed to be inputs for the model:

- Calendar month/year to be scheduled
- Pre-scheduled days for training classes
- Training resources including classrooms sizes and availability for trainees
- Number of instructors and/or evaluators required for each training event
- Number of slots per day available for TR and Evaluations
- Number of personnel and their job titles, roles, crewmate, certifications and squadron they belong to
- Unavailable days for personnel due to Leave, Duty Not Including Alert (DNIA), Temporary Duty (TDY), etc.

Sets

Table 4 lists the main sets, and their indices, that are defined under the Model Data section of the AFOCS model. However, more sets that are in the model will be defined later in this document:

Table 4: Sets of the AFOCS model

Set	Index
classes	cl
day	d, d2, the day of the month (i.e. 15)
dayOfweek	dw, the day of the week = {Su, M, T, W, Th, F, S}
events	e, any event = {O-day,B-day Alert, TR, T1, T3, T4, 'T3/T4', eval, Backup}
simulationEvents	simEvent, this is a subset of the set "e" = {TR, eval}
trainingEvents	train, this is a subset of the set "e" = {TR, T1, T3, T4, 'T3/T4'}
classroomEvents	class, this is a subset of the set "train" = {T1, T3, T4, 'T3/T4'}
job	j,
month	m, the month = {1,2,….,12}
personnel	p, p2, the personnel's name (Input through excel)
crewdogs	cw, this is a subset of the set "p"
instructors	ins, this is a subset of the set "p"
evaluators	evals, this is a subset of the set "p"
flightcommanders	fc, this is a subset of the set "p"
position	pos, the position of the personnel (CDR, DEP)
sites	site, all 15 sites = {A,B, C, D, E-SCP, F, G, H, I, J-SCP, K, L, M, N, O-SCP}
scpsites	scp, this is a subset of the set "site" and it represent the SCP sites = {E-SCP, J-SCP, O-SCP}
slots	sl, sl2,
simulators	sim,
squadrons	sq, the three main squadrons = {Squadron1, Squadron2, Squadron3}
year	y
numOfAlerts	n, = {1, 2, 3, 4, 5, 6, 7, 8}

Parameters

Table 5 lists all the parameters, and their meaning, of the model.

12.2.2 Model Variables

Table 6 lists all the variables, and their definition, of the model.

AIMMS allows us to define sets that represent multiple variables or constraints, and that will allow to organize and formulate long mathematical programs. In this section of the model, eight sets were defined:

AlertVariables	= {V1, V17, V18, V22, V23, V27, V29, V34, V39, V53. V54, V55, V57}
TrainingVariables	= {V2-V9, V17-V20, V22-V25, V27, V30-V32, V35-V37, V40-V42, V53, V54, V56, V58}
SlotVariables	= {V10-V20, V22-V25, V27, V53, V54, V58}
CrewIntegrityVariables	= {V17-V20, V22, V24, V25, V27, V54}
BackupCrewVariables	= {V21, V26, V28, V33, V38, V43, V53, V54, V56, V58}
DynamicAlertVariables	= {V1-V6, V17, V18, V22, V23, V27, V29, V34, V39, V44, V45, V52-V55, V57}
DynamicTrainingVariables	= {V2-V15, V17-V28, V30-V33, V35-V38, V40-V44, V46-V54, V56, V58}
DynamicSlotVariables	= {V10-V20, V22-V25, V27, V44, V52, V53, V54, V58 }

12.2.3 Model Constraints

Table 7 lists all the constraints, and their meaning, for this problem.

In this section of the model, the following sets were defined:

AlertConstraints	={V22, V23, V27, V34, V54, V55, V57, C1-C11, C67, C68, C80, C81}
TrainingConstraints	={V9, V22-V25, V27, V30-V32, V35-V37, V54, V56, V58, C5, C12-C41, C67-C70, C82-C87}
SlotConstraints	={V22-V25, V27, V54, V58, C5, C12, C13, C26, C42-C71}
CrewIntegrityConstraints	={V22, V24, V25, V27, V54, C67-C71}
CrewIntegrityConstraints	={V26, V33, V38, V54, V56, V58, C5, C12, C13, C60, C61, C72-79, C88, C89}
DynamicAlertConstraints	={V22, V23, V27, V34, V52, V54, V55, V57, C1-C11, C67, C68, C80, C81, C90-C96, C103-C106}
DynamicTrainingConstraints	={V9, V22-V27, V33, V38, 52, 54, C5, C12-C41, C67-C70, C72-C79, C88-C104, C107-C117}
DynamicSlotConstraints	={V22-V25, V27, V52, V54, C12, C13, C26, C42-C71, C90-C102}

12.2.4 Model Output Parameters

Table 8 lists all the output parameters, and their meaning, for this problem.

Those are the parameters that will be the output of the model execution. All those parameters represent the schedule for all events (shifts and training).

Table 5: Parameters of the AFOCS model

Parameter	Meaning	P #
	Raw Data	
$\mathbf{NumDaysInMonth_{m,y}}$	= Number of days in month 'm' year 'y'	(P1)
$\mathbf{FirstDayOfMonthDW_{m,y,dw}}$	$= \begin{cases} 1, \text{first day of month 'm' in year 'y' day of the week 'dw'} \\ 0, \text{else} \end{cases}$	(P2)
$\mathbf{DWtoDay_{d,dw}}$	$= \begin{cases} 1, \text{day of week 'dw' is day 'd'} \\ 0, \text{else} \end{cases}$	(P3)
InputMonth	= The month of the schedule	(P4)
InputYear	= The year of the schedule	(P5)
$\mathbf{SiteToSquadron_{site,sq}}$	$= \begin{cases} 1, \text{site 'site' assigned to Squadron 'sq'} \\ 0, \text{else} \end{cases}$	(P6)
$\mathbf{PersToSquadron_{p,sq}}$	$= \begin{cases} 1, \text{personnel 'p' belongs to Squadron 'sq'} \\ 0, \text{else} \end{cases}$	(P7)
$\mathbf{PersonSquadron_{p}}$	= This is a string parameter, and it represent the squadron personnel 'p' belongs to	(P8)
$\mathbf{PersToPos_{p,pos}}$	$= \begin{cases} 1, \text{personnel 'p' has position 'pos'} \\ 0, \text{else} \end{cases}$	(P9)
$\mathbf{PersonPosition_{p}}$	= This is a string parameter, and it represent personnel 'p' position	(P10)
$\mathbf{PersToJob_{p,j}}$	$= \begin{cases} 1, \text{personnel 'p' has job 'j'} \\ 0, \text{else} \end{cases}$	(P11)
$\mathbf{PersAlertAvailDays_{p,d}}$	$= \begin{cases} 1, \text{personnel 'p' available on day 'd'} \\ 0, \text{else} \end{cases}$	(P12)
$\mathbf{PersTrainAvailDays_{p,d}}$	$= \begin{cases} 1, \text{personnel 'p' available on day 'd'} \\ 0, \text{else} \end{cases}$	(P13)
$\mathbf{CrewMate_{p}}$	= This is a string parameter, and it represent personnel 'p' crewmate	(P14)

Parameter	Meaning	P #
CrewMates $_{p,p2}$	$= \begin{cases} 1, \text{personnel 'p' is crewed with personnel 'p2'} \\ 0, \text{else} \end{cases}$	(P15)
PersSCPCertified $_{p}$	$= \begin{cases} 1, \text{personnel 'p' is SCP certified} \\ 0, \text{else} \end{cases}$	(P16)
PersClassTrainingReq $_{p,class}$	$= \begin{cases} 1, \text{personnel 'p' has to attend training class 'class'} \\ 0, \text{else} \end{cases}$	(P17)
PersSimEventReq $_{p,simEvent}$	$= \begin{cases} 1, \text{personnel 'p' has to attend simulation event 'simEvent'} \\ 0, \text{else} \end{cases}$	(P18)
MaxAlertsPerJob$_{j}$	= a range between $0 - 8$ which represent the maximum number of alerts per job	(P19)
AlertReq $_{site,d,pos}$	$= \begin{cases} 1, \text{site 'site'day 'd' position 'pos' is an alert required} \\ 0, \text{else} \end{cases}$	(P20)
User Input		
ClassroomSize$_{class}$	= classroom size for each training event, always >= 0	(P21)
NumberOfSlotsForTraining	= number of slots per day available for training (i. e. 5)	(P22)
DayTrainingClassHeld $_{d,class}$	$= \begin{cases} 1, \text{day 'd'is held fot training class room event 'class'} \\ 0, \text{else} \end{cases}$	(P23)
DaysSimsAvailable $_{d}$	$= \begin{cases} 1, \text{the simulator is available on day 'd'} \\ 0, \text{else} \end{cases}$	(P24)
NumberOfInstructoresRequired $_{d,train}$	= Number of instructors required for day 'd' for trainingEvent 'train'	(P25)
NumberOfEvaluatoresRequired $_{d,simEvent}$	= Number of evaluators required for day 'd' for 'simEvent'	(P26)
SlotsAvailable $_{sl,d}$	$= \begin{cases} 1, \text{slot 'sl' available on day 'd'} \\ 0, \text{else} \end{cases}$	(P27)
SimsAvailable $_{sim,d}$	$= \begin{cases} 1, \text{simulators 'sim' available on day 'd'} \\ 0, \text{else} \end{cases}$	(P28)
Dynamic Portion Input		

Parameter	Meaning	P #
ProblemDay	= The day that the problem occured (resheduleing a shift)	(P29)
PersonNowUnavailable	= The personnel who is now unavailable	(P30)
NewPersonAlertAvailability $_{p,d}$	$= \begin{cases} 1, \text{personnel 'p' available for alert on day 'd'} \\ 0, \text{else} \end{cases}$	(P31)
NewPersonTrainAvailability $_{p,d}$	$= \begin{cases} 1, \text{personnel 'p' available for training on day 'd'} \\ 0, \text{else} \end{cases}$	(P32)
CompletedSchedule$_{p,d,e\ (d \le P32-1)}$	$= \begin{cases} 1, \text{personnel 'p' scheduled event 'e' on day 'd'} \\ 0, \text{else} \end{cases}$	(P33)
CompletedAlertAssignments$_{p,site,d\ (d \le P32-1)}$	$= \begin{cases} 1, \text{personnel 'p' scheduled site 'site' on day 'd'} \\ 0, \text{else} \end{cases}$	(P34)
CompletedClassTrainingAssignments$_{p,d,class\ (d \le P32-1)}$		(P35)
	$= \begin{cases} 1, \text{personnel 'p' scheduled class 'class' on day 'd'} \\ 0, \text{else} \end{cases}$	(P36)
CompletedSimTrainingAssignments$_{p,d,simEvent\ (d \le P32-1)}$		(P37)
	$= \begin{cases} 1, \text{personnel 'p' scheduled simulation 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(P38)
CompletedInsTrainAssignments$_{ins,d,train\ (d \le P32-1)}$		(P39)
	$= \begin{cases} 1, \text{personnel 'ins' scheduled training 'train' on day 'd'} \\ 0, \text{else} \end{cases}$	(P40)
CompletedEvalTrainAssignments$_{evals,d,simEvent(d \le P32-1)}$		(P41)
	$= \begin{cases} 1, \text{personnel 'evals' scheduled training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(P42)
CompletedAssignPersonToSlot$_{sl,sim,d,p}$	$= \begin{cases} 1, \text{personnel 'p' assigned simulators 'sim' to slot 'sl' on day 'd'} \\ 0, \text{else} \end{cases}$	(P43)
CompletedEvalToSlot$_{sl,sim,d,evals}$	$= \begin{cases} 1, \text{personnel 'evals' assigned simulators 'sim' to slot 'sl' on day 'd'} \\ 0, \text{else} \end{cases}$	(P44)

Parameter	Meaning	P #
$\mathbf{CompletedInsToSlot_{sl,sim,d,ins}}$	$= \begin{cases} 1, \text{personnel 'ins' assigned simulators 'sim' to slot 'sl' on day 'd'} \\ 0, \text{else} \end{cases}$	(P45)
$\mathbf{CompletedSimToSlot_{sl,d,sim}}$	$= \begin{cases} 1, \text{assigned simulators 'sim' to slot 'sl' on day 'd'} \\ 0, \text{else} \end{cases}$	(P46)
$\mathbf{CompletedNumSlotsPerIns_{d,ins}}$	$= \begin{cases} 1, \text{assigned instructors 'ins' on day 'd'} \\ 0, \text{else} \end{cases}$	(P47)
$\mathbf{CompletedSlotForSimEvent_{sl,sim,d,simEvent}}$	$= \begin{cases} 1, \text{assigned slot 'sl' simulator 'sim' training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(P48)
$\mathbf{CompletedBackupCrew_{p,d}}$	$= \begin{cases} 1, \text{assigned backup crew personnel 'p' on day 'd'} \\ 0, \text{else} \end{cases}$	(P49)
Other		
$\mathbf{PersAlertSchedule_{p,d,e}}$	$= \begin{cases} 1, \text{personnel 'p' schedule for event 'e' on day 'd'} \\ 0, \text{else} \end{cases}$	(P50)
$\mathbf{PersnSimEventSchedule_{p,d,simEvent}}$	$= \begin{cases} 1, \text{personnel 'p' on day 'd' schedule for simulation training 'simEvent'} \\ 0, \text{else} \end{cases}$	(P51)
$\mathbf{PersnClassSchedule_{p,d,class}}$	$= \begin{cases} 1, \text{personnel 'p' on day 'd' schedule for class room event 'class'} \\ 0, \text{else} \end{cases}$	(P52)
$\mathbf{PersBackupsSchedule_{p,d,e}}$	$= \begin{cases} 1, \text{backup personnel 'p' on day 'd' schedule for event 'e'} \\ 0, \text{else} \end{cases}$	(P53)

Table 6: Variables of the AFOCS model

Variables	Meaning	V #
	Alert Variables	
$\mathbf{AssignPersToSite}_{\mathbf{p,site,d}}$	$= \begin{cases} 1, \text{personnel 'p' is assigned in site 'site' for day 'd'} \\ 0, \text{else} \end{cases}$	(V1)
	Training Variables	
$\mathbf{AssignPersonToTrainingClass}_{\mathbf{p,d,class}}$	$= \begin{cases} 1, \text{personnel 'p' assigned for training classroom 'class' on day 'd'} \\ 0, \text{else} \end{cases}$	(V2)
$\mathbf{AssignPersonToSimTraining}_{\mathbf{p,d,SimEvent}}$	$= \begin{cases} 1, \text{personnel 'p' assigned for simulator training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(V3)
$\mathbf{AssignInstructorsToTrainClass}_{\mathbf{ins,d,class}}$	$= \begin{cases} 1, \text{instructor 'ins' assigned to instruct training classroom 'class' on day 'd'} \\ 0, \text{else} \end{cases}$	(V4)
$\mathbf{AssignInstructorsToTrainTRs}_{\mathbf{ins,d,SimEvent}}$	$= \begin{cases} 1, \text{instructor 'ins' assigned to instruct simulator training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(V5)
$\mathbf{AssignEvaluatorsToEvals}_{\mathbf{evals,d,SimEvent}}$	$= \begin{cases} 1, \text{evaluator 'evals' assigned to evaluate simulator training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(V6)
$\mathbf{ClassroomTrainingCompleted}_{\mathbf{p,class}}$	$= \begin{cases} 1, \text{personnel 'p' completed class room training 'class'} \\ 0, \text{else} \end{cases}$	(V7)
$\mathbf{ClassRequirement}_{\mathbf{p,class\ (class<>\frac{T3}{T4})}}$	$= \begin{cases} 1, \text{personnel 'p'required to attend class room training 'class' not including} \frac{T3}{T4} \\ 0, \text{else} \end{cases}$	(V8)
PeopleAssignedToTraining	= Total number of people assigned for training	(V9)
	Slot Variables	
$\mathbf{AssignPersonToSlot}_{\mathbf{sl,sim,d,p}}$	$= \begin{cases} 1, \text{personnel 'p' assigned to slot 'sl'simulator 'sim'on day 'd'} \\ 0, \text{else} \end{cases}$	(V10)
$\mathbf{AssignEvaluatorsToSlot}_{\mathbf{sl,sim,d,evals}}$	$= \begin{cases} 1, \text{evaluator 'evals' assigned to slot 'sl'simulator 'sim'on day 'd'} \\ 0, \text{else} \end{cases}$	(V11)
$\mathbf{AssignInstructorsToSlot}_{\mathbf{sl,sim,d,ins}}$	$= \begin{cases} 1, \text{instructor 'ins' assigned to slot 'sl'simulator 'sim'on day 'd'} \\ 0, \text{else} \end{cases}$	(V12)

Variables	Meaning	V #
AssignSimulatorToSlot $_{sl,d,sim}$	$= \begin{cases} 1, \text{assign simulator 'sim' to slot 'sl' on day 'd'} \\ 0, \text{else} \end{cases}$	(V13)
NumberSlotsPerInstructor $_{d,ins}$	= only assign instructors to 2 slots per day = {0,1,2}	(V14)
SlotUsedForSimEvent $_{sl,sim,d,simEvent}$	$= \begin{cases} 1, \text{slot 'sl' of simulaotr 'sim' being used for 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(V15)
SlotsAssigned	= Total number of slots assigned	(V16)
	Crew Integrity Variables	
PersPairedWithMateEvent $_{p,d,e}$	$= \begin{cases} 1, \text{personnel 'p' is paired with the assigned mate for event 'e' on day 'd'} \\ 0, \text{else} \end{cases}$	(V17)
PersPairedWithMateSite $_{p,d,site}$	$= \begin{cases} 1, \text{personnel 'p' is paired with the assigned mate on site 'site' on day 'd'} \\ 0, \text{else} \end{cases}$	(V18)
PersPairedWithMateClass $_{p,d,class}$	$= \begin{cases} 1, \text{personnel 'p' is paired with the assigned mate for training class 'class' on day 'd'} \\ 0, \text{else} \end{cases}$	(V19)
PersPairedWithMateSimEvent $_{p,d,simEvent}$	$= \begin{cases} 1, \text{personnel 'p' is paired with the assigned mate for training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	(V20)
PersPairedWithMateBackup$_{p,d}$	$= \begin{cases} 1, \text{personnel 'p' is paired with the assigned mate on day 'd' as a backup crew} \\ 0, \text{else} \end{cases}$	(V21)
PersonCrewIntegrityAlert $_{p}$	$= \sum_{d,site} V18$	(V22)
PersonCrewIntegrityEvent $_{p}$	$= \sum_{d,e} V17$	(V23)
PersonCrewIntegrityClass $_{p}$	$= \sum_{d,class} V19$	(V24)
PersonCrewIntegritySimEvent$_{p}$	$= \sum_{d,simEvent} V20$	(V25)
PersonCrewIntegrityBackup$_{p}$	$= \sum_{d} V21$	(V26)

Variables	Meaning	V #
PersonCrewIntegrityAllEvents$_p$	$= V22 + V24 + V25$	(V27)
	Backup Crew Variables	
PersAssignedAsBackupCrew$_{p,d}$	$= \begin{cases} 1, \text{personnel 'p' assigned for day 'd'} \\ 0, \text{else} \end{cases}$	(V28)
	Different Days Penalty Variables	
NumDaysPersonOnAlert $_p$	= Number of days personnel 'p' is on alert	(V29)
NumDaysInsTeachingClass$_{ins}$	$= \sum_{d,class} \text{AssignInstructorsToTrainClass}_{ins,d,class}$, $\left(\sum_{d,class} V4\right)$	(V30)
NumDaysInsTeachingTRs$_{ins}$	$= \sum_{d} \text{AssignInstructorsToTrainTRs}_{ins,d,'TR'}$	(V31)
NumDaysEvalsTeachingEvals$_{evals}$	$= \sum_{d} \text{AssignEvaluatorsToEvals}_{evals,d,'eval'}$	(V32)
NumdaysPersonOnBackup$_p$	$= \sum_{d} \text{PersAssignedAsBackupCrew}_{p,d}$, $\left(\sum_{d} V28\right)$	(V33)
AverageDaysOnAlertJob$_j$	$= \frac{\sum_p(\text{NumDaysPersonOnAlert}_p * \text{PersToJob}_{p,j})}{\sum_p \text{PersToJob}_{p,j}}$, $\left(\frac{\sum_p(V29 + P11)}{\sum_p P11}\right)$	(V34)
AverageDaysInsTeachingClass	$= \frac{\sum_{ins}(\text{NumDaysInsTeachingClass}_{ins})}{\sum_{ins} \text{PersToJob}_{ins,\ 'instructor'}}$	(V35)
AverageDaysInsTeachingTR	$= \frac{\sum_{ins}(\text{NumDaysInsTeachingTRs}_{ins})}{\sum_{ins} \text{PersToJob}_{ins,\ 'instructor'}}$	(V36)
AverageDaysEvalsTeachingEvals	$= \frac{\sum_{evals}(\text{NumDaysEvalsTeachingEvals}_{evals})}{\sum_{evals} \text{PersToJob}_{evals,\ 'evaluator'}}$	(V37)
AverageDaysOnBackup $_j$	$= \frac{\sum_p(\text{NumdaysPersonOnBackup}_p * \text{PersToJob}_{p,j})}{\sum_p \text{PersToJob}_{p,j}}$, $\left(\frac{\sum_p(V33 + P11)}{\sum_p P11}\right)$	(V38)
DiffDaysOnAlert$_p$	Free	(V39)

Variables	Meaning	V #
$\mathbf{DiffDaysInsTeachingClass_{ins}}$	Free	(V40)
$\mathbf{DiffDaysInsTeachingTR_{ins}}$	Free	(V41)
$\mathbf{DiffDaysEvalsTeachEvals_{evals}}$	Free	(V42)
$\mathbf{DiffDaysOnBackup_{p}}$	Free	(V43)
Dynamic Portion Variables		
$\mathbf{DiffOrigSchedule_{p,d,e}}$	Free	(V44)
$\mathbf{DiffOrigAlertAssignments_{p,site,d}}$	Free	(V45)
$\mathbf{DiffOrigClassTrainingAssignments_{p,d,class}}$	Free	(V46)
$\mathbf{DiffOrigSimTrainingAssignment_{p,d,simEvent}}$	Free	(V47)
$\mathbf{DiffOrigInsTrainClassAssignments_{ins,d,class}}$	Free	(V48)
$\mathbf{DiffOrigInsTrainTRAssignments_{ins,d,simEvent}}$	Free	(V49)
$\mathbf{DiffOrigEvalTrainAssignments_{evals,d,simEvent}}$	Free	(V50)
$\mathbf{DiffOrigBackupCrew_{p,d}}$	Free	(V51)
TotalDiffFromEntireOrigSchedule	$= \sum_{p,d,e,site} V44 + V45 + V46 + V47 + V48 + V49 + V50 + V51$	(V52)
Global Variables		
$\mathbf{PersonSchedule}_{p,d,e}$	$= \begin{cases} 1, \text{personnel 'p' on day 'd' is scheduled for event 'e'} \\ 0, \text{else} \end{cases}$	(V53)

Variables	Meaning	V #
IntegralCrewRate	$\sum_{p} \text{PersonCrewIntegrityAllEvents}_{p}$, $\left(\sum_{p} \text{V27}\right)$	(V54)
TotalDiffDaysAlert	$\sum_{p} \text{DiffDaysOnAlert}_{p}$, $\left(\sum_{p} \text{V39}\right)$	(V55)
TotalDiffDaysTraining	$\sum_{ins} (\text{DiffDaysInsTeachingClass}_{ins} + \text{DiffDaysInsTeachingTR}_{ins}) + \sum_{evals} (\text{DiffDaysEvalsTeachEvals}_{evals}$, $\left(\sum_{ins} (\text{V41} + \text{V42}) + \sum_{evals} \text{V43}\right)$	(V56)
ObjectiveFunctionAlert	IntegralCrewRate – TotalDiffDaysAlert , $(V54 - V55)$	(V57)
ObjectiveFunctionTraining	IntegralCrewRate – TotalDiffDaysTraining , $(V54 - V56)$	(V58)

Table 7: Constraints of the AFOCS model

Meaning	Constraints		C#
Alert Constraints			
To satisfy all the alert requirements (to schedule all the alert shifts)	$\sum_{p} \text{AssignPersToSite}_{p,site,d} * \text{PersToPos}_{p,pos} * \text{PersAlertAvailDays}_{p,d} = \text{AlertReq}_{site,d,pos}$	$\forall$ site, d, pos	(C1)
To assign 2 personnel per site	$\sum_{p} \text{AssignPersToSite}_{p,site,d} = 2$	$\forall$ site, d	(C2)
Assign person to site within his squadron	$\text{AssignPersToSite}_{p,site,d} * \text{SiteToSquadron}_{site,sq} \leq \text{PersToSquadron}_{p,sq}$	$\forall$ p, d, site, sq	(C3)
Assign person to no more than one site	$\sum_{site} \text{AssignPersToSite}_{p,site,d} \leq 1$	$\forall$ p, d	(C4)
Assign person to one event (i.e. Alert, O'day or training)	$\sum_{e} \text{PersonSchedule}_{p,d,e} \leq 1$	$\forall$ p, d	(C5)
To not assign any shifts more than the max. number of shifts per month for personnel 'p'	$\sum_{d} \text{PersonSchedule}_{p,d,'Alert'} * \text{PersToJob}_{p,j} \leq \text{MaxAlertsPerJob}_{j}$	$\forall$ p, j	(C6)
To assign SCP certified personal for SCP sites)	$\text{AssignPersToSite}_{p,scp,d} \leq \text{PersSCPCertified}_{p}$	$\forall$ p, d, scp	(C7)
Set person alert shifts (schedule)	$\sum_{site} \text{AssignPersToSite}_{p,site,d} = \text{PersonSchedule}_{p,d,'Alert'}$	$\forall$ p, d	(C8)
To set an O'day after an alert day	$\sum_{site} \text{AssignPersToSite}_{p,site,d} = \text{PersonSchedule}_{p,d2,\ 'O-day'}$	$\forall$ p, d, d2 | d2 = d + 1	(C9)
To not schedule the next event until at least 3 days	$\sum_{site} \text{AssignPersToSite}_{p,site,d} = \text{PersonSchedule}_{p,d2,\ 'B-day'}$	$\forall$ p, d, d2 | d2 = d + 2	(C10)

Meaning	Constraints		C#
The total number of days person is on alert	$\sum_{d} \text{PersonSchedule}_{p,d,'Alert'} = \text{NumDaysPersonOnAlert}_{p}$	$\forall$ p	(C11)
Training Constraints			
Save the dates the person is on alert	$\text{PersonSchedule}_{p,d,'Alert'} = \text{PersAlertSchedule}_{p,d,'Alert'}$	$\forall$ p, d	(C12)
Save the dates the person is off (O'day)	$\text{PersonSchedule}_{p,d,'O-day'} = \text{PersAlertSchedule}_{p,d,'O-day'}$	$\forall$ p, d	(C13)
To meet number of evaluators requirement on days that personnel are taking their annual evaluation	$\sum_{evals} (\text{AssignEvaluatorsToEvals}_{evals,d,'eval'} * \text{PersTrainAvailDays}_{evals,d}) = \text{NumberOfEvaluatoresRequired}_{d,'eval'} * \text{DaysSimsAvailable}_{d} * \sum_{p} \text{AssignPersonToSimTraining}_{p,d,'eval'}$	$\forall$ d	(C14)
To satisfy training class requirement if the person is already assigned to a training on any day of the month.	$\sum_{d} (\text{AssignPersonToTrainingClass}_{p,d,class} * \text{PersTrainAvailDays}_{p,d} * \text{DayTrainingClassHeld}_{d,class}) = \text{ClassroomTrainingCompleted}_{p,class}$	$\forall$ p, class	(C15)
To meet number instructors required for TR training	$\text{NumberOfInstructoresRequired}_{d,'TR'} * \sum_{sim} \text{SimsAvailable}_{sim,d} * \sum_{sl} \text{SlotsAvailable}_{sl,d} = \sum_{ins} \text{AssignInstructorsToTrainTRs}_{ins,d,'TR'} * \text{PersTrainAvailDays}_{ins,d}$	$\forall$ d	(C16)
To meet number of instructors required for class training	$\sum_{ins} (\text{AssignInstructorsToTrainClass}_{ins,d,class} * \text{PersTrainAvailDays}_{ins,d}) = \text{NumberOfInstructoresRequired}_{d,class} * \text{DayTrainingClassHeld}_{d,class}$	$\forall$ d, class	(C17)
To meet personnel required simulation training	$\sum_{d} (\text{AssignPersonToSimTraining}_{p,d,SimEvent} * \text{PersTrainAvailDays}_{p,d}) = \text{PersSimEventReq}_{p,simEvent}$	$\forall$ p, simEvent	(C18)

Meaning	Constraints		C#
T assign the same number of commanders and deputies to the simulation training 'simEvent'	$\sum_{p}(\text{AssignPersonToSimTraining}_{p,d,\text{SimEvent}} * \text{PersToPos}_{p,\text{'Deputy'}}) = \sum_{p}(\text{AssignPersonToSimTraining}_{p,d,\text{SimEvent}} * \text{PersToPos}_{p,\text{'Commander'}})$	∀ d, simEvent	(C19)
To meet all the class training requirements	$\text{ClassRequirement}_{p,class} \geq \text{PersClassTrainingReq}_{p,class}$	∀ p, class (class <> T3/T4)	(C20)
Total number of personnel attending a training class do not exceed the size of the class	$\sum_{p}\text{AssignPersonToTrainingClass}_{p,d,class} \leq \text{ClassroomSize}_{class} * \text{DayTrainingClassHeld}_{d,class}$	∀ d, class	(C21)
Total number of personnel attending a simulation training do not exceed the available number of slots	$\sum_{p,simEvent}\text{AssignPersonToSimTraining}_{p,d,\text{SimEvent}} \leq 2 * \sum_{sim}\text{SimsAvailable}_{sim,d} * \sum_{sl}\text{SlotsAvailable}_{sl,d}$	∀ d	(C22)
To assign 2 personnel (crew members who have alert the next day) per simulator for the available slots (3)	$\sum_{p,simEvent}(\text{AssignPersonToSimTraining}_{p,d,\text{SimEvent}} * \text{PersAlertSchedule}_{p,d+1,\text{'Alert'}}) \leq 6\sum_{sim}\text{SimsAvailable}_{sim,d}$	∀ d	(C23)
To assign 2 personnel (instructors)for each slot (3) to instruct the simulator training	$\sum_{ins}(\text{AssignInstructorsToTrainTRs}_{ins,d,\text{'TR'}} * \text{PersAlertSchedule}_{ins,d+1,\text{'Alert'}}) \leq 6\sum_{sim}\text{SimsAvailable}_{sim,d}$	∀ d	(C24)
To assign 3 personnel (evaluators) for each evaluation slot (3) and not to include the instructors who has an alert shift next day	$\sum_{evals}\text{AssignEvaluatorsToEvals}_{evals,d,\text{'eval'}} * \text{PersAlertSchedule}_{evals,d+1,\text{'Alert'}} \leq 9\sum_{sim}\text{SimsAvailable}_{sim,d} - \sum_{ins}(\text{AssignInstructorsToTrainTRs}_{ins,d,\text{'TR'}} * \text{PersAlertSchedule}_{ins,d+1,\text{'Alert'}})$	∀ d	(C25)
To limit the instructors to the annual evaluation training	$\text{PersonSchedule}_{ins,d,\text{'eval'}} \leq \text{PersSimEventReq}_{ins,\text{'eval'}}$	∀ ins, d	(C26)

Meaning	Constraints		C#
Personnel that complete T3/T4 training class, they satisfied the T3 class requirement	$\text{ClassroomTrainingCompleted}_{p,'T3/T4'} + \text{ClassroomTrainingCompleted}_{p,'T3'} = \text{ClassRequirement}_{p,'T3'}$	$\forall$ p	(C27)
Personnel that complete T3/T4 training class, they satisfied the T4 class requirement	$\text{ClassroomTrainingCompleted}_{p,'T3/T4'} + \text{ClassroomTrainingCompleted}_{p,'T4'} = \text{ClassRequirement}_{p,'T4'}$	$\forall$ p	(C28)
Personnel that complete T1 training class, they satisfied the T1 class requirement	$\text{ClassroomTrainingCompleted}_{p,'T1'} = \text{ClassRequirement}_{p,'T1'}$	$\forall$ p	(C29)
To set crew members simulator event training schedule	$\text{AssignPersonToSimTraining}_{cw,d,SimEvent} = \text{PersonSchedule}_{cw,d,simEvent}$	$\forall$ cw, d, simEvent	(C30)
To set crew members training classes schedule	$\text{PersonSchedule}_{cw,d,class} = \text{AssignPersonToTrainingClass}_{cw,d,class}$	$\forall$ cw, d, class	(C31)
To set evaluators schedule for the annual training evaluation	$\text{AssignEvaluatorsToEvals}_{evals,d,'eval'} + \text{AssignPersonToSimTraining}_{evals,d,'eval'} = \text{PersonSchedule}_{evals,d,'eval'}$	$\forall$ evals, d	(C32)
To set evaluators schedule for the TR training evaluation	$\text{AssignPersonTosimTrainin}_{evals,d,'TR'} = \text{PersonSchedule}_{evals,d,'TR'}$	$\forall$ evals, d	(C33)
To set evaluators training classes schedule	$\text{PersonSchedule}_{evals,d,class} = \text{AssignPersonToTrainingClass}_{evals,d,class}$	$\forall$ evals, d, class	(C34)
To set instructors training classes schedule	$\text{AssignInstructorsToTrainClass}_{ins,d,class} + \text{AssignPersonToTrainingClass}_{ins,d,class} = \text{PersonSchedule}_{ins,d,class}$	$\forall$ ins, d, class	(C35)
To set instructors TR training schedule	$\text{AssignInstructorsToTrainTRs}_{ins,d,'TR'} = \text{PersonSchedule}_{ins,d,'TR'}$	$\forall$ ins, d	(C36)
To set flight commander training classes schedule	$\text{PersonSchedule}_{fc,d,class} = \text{AssignPersonToTrainingClass}_{fc,d,class}$	$\forall$ fc, d, class	(C37)
To set flight commander simulation event training schedule	$\text{AssignPersonToSimTraining}_{fc,d,SimEvent} = \text{PersonSchedule}_{fc,d,simEvent}$	$\forall$ fc, d, simEvent	(C38)

Meaning	Constraints		C#
To set instructor to either teach or attend a training class	$\text{AssignPersonToTrainingClass}_{\text{ins,d,class}} + \text{AssignInstructorsToTrainClass}_{\text{ins,d,class}} \leq 1$	$\forall$ ins, d, class	(C39)
To set instructor to either teach or attend a TR training class	$\text{AssignPersonToSimTraining}_{\text{ins,d,'TR'}} + \text{AssignInstructorsToTrainTRs}_{\text{ins,d,'TR'}} \leq 1$	$\forall$ ins, d	(C40)
To set evaluators to either teach or attend a simulation event training class	$\text{AssignEvaluatorsToEvals}_{\text{evals,d,simEvent}} + \text{AssignPersonToSimTraining}_{\text{evals,d,SimEvent}} \leq 1$	$\forall$ evals, d, simEvent	(C41)
Slot Constraints			
If a person is assigned to a simulation event training on day 'd', then they must be assigned to a slot.	$\sum_{\text{sl,sim}} \text{AssignPersonToSlot}_{\text{sl,sim,d,p}} = \sum_{\text{simEvent}} \text{AssignPersonToSimTraining}_{\text{p,d,SimEvent}}$	$\forall$ d, p	(C42)
No more than one commander get assigned to a slot if available	$\sum_{\text{p}} (\text{AssignPersonToSlot}_{\text{sl,sim,d,p}} * \text{PersToPos}_{\text{p,'Commander'}}) \leq \text{SlotsAvailable}_{\text{sl,d}} * \text{SimAvailable}_{\text{sim,d}}$	$\forall$ sl, sim, d	(C43)
No more than one deputy get assigned to a slot if available	$\sum_{\text{p}} (\text{AssignPersonToSlot}_{\text{sl,sim,d,p}} * \text{PersToPos}_{\text{p,'Deputy'}}) \leq \text{SlotsAvailable}_{\text{sl,d}} * \text{SimAvailable}_{\text{sim,d}}$	$\forall$ sl, sim, d	(C44)
To assign personnel 'p' to no more than one slot	$\sum_{\text{sl,sim}} \text{AssignPersonToSlot}_{\text{sl,sim,d,p}} \leq 1$	$\forall$ d, p	(C45)
To assign simulators to the slots (if the simulators are available)	$\text{AssignSimulatorToSlot}_{\text{sl,d,sim}} * \text{SimAvailable}_{\text{sim,d}} \geq 1 * \text{SlotsAvailable}_{\text{sl,d}}$	$\forall$ sl, d, sim	(C46)
To assign people to a slot that has the same simulation event training as the Slot. At most 2 people can be assigned to each simulator	$\sum_{\text{p}} (\text{AssignPersonToSlot}_{\text{sl,sim,d,p}} * \text{PersonSimEventSchedule}_{\text{p,d,simEvent}}) \leq 2 * \text{SlotUsedForSimEvent}_{\text{sl,sim,d,simEvent}}$	$\forall$ sim, sl, d, simEvent	(C47)
To set 2 instructors per TR	$\sum_{\text{ins}} \text{AssignInstructorsToSlot}_{\text{sl,sim,d,ins}} = 2 * \text{SlotUsedForSimEvent}_{\text{sl,sim,d,'TR'}}$	$\forall$ sl, sim, d	(C48)

Meaning	Constraints		C#
Limits the number of slots Instructors can do to 2 slots, and only allows Instructors assigned to do TRs on that day to be assigned to slots	$\sum_{sl,sim} \text{AssignInstructorsToSlot}_{sl,sim,d,ins} \leq 2 * \text{AssignInstructorsToTrainTRs}_{ins,d,'TR'}$	$\forall$ d, ins	(C49)
To assign instructors to no more than 1 simulator in a given slot	$\sum_{sim} \text{AssignInstructorsToSlot}_{sl,sim,d,ins} \leq 1$	$\forall$ d, ins, sl	(C50)
To assign evaluators to no more than 1 simulator in a given slot	$\sum_{sim} \text{AssignEvaluatorsToSlot}_{sl,sim,d,evals} \leq 1$	$\forall$ d, evals, sl	(C51)
To set the number of slots per instructor	$\sum_{sl,sim} \text{AssignInstructorsToSlot}_{sl,sim,d,ins} = \text{NumberSlotsPerInstructor}_{d,ins}$	$\forall$ d, ins	(C52)
To set instructors to either teach or attend a TR slot	$\text{AssignInstructorsToSlot}_{sl,sim,d,ins} + \text{AssignPersonToSlot}_{sl,sim,d,ins} \leq 1,$	$\forall$ sl, sim, d, ins	(C53)
To set 3 evaluators for the annual evaluation training	$\sum_{evals} \text{AssignEvaluatorsToSlot}_{sl,sim,d,evals} = 3 * \sum_{p} (\text{AssignPersonToSlot}_{sl,sim,d,p} * \text{PersonSimEventSchedule}_{p,d,'eval'})$	$\forall$ sl, sim, d	(C54)
To set evaluators to either teach or attend the annual evaluation training	$\text{AssignEvaluatorsToSlot}_{sl,sim,d,evals} + \text{AssignPersonToSlot}_{sl,sim,d,evals} \leq 1$	$\forall$ sl, sim, d, evals	(C55)
To make sure that slots can only be used for 1 event (TR or eval)	$\text{SlotUsedForSimEvent}_{sl,sim,d,'TR'} + \text{SlotUsedForSimEvents}_{sl,sim,d,'eval'} \leq 1$	$\forall$ sim, sl, d	(C56)
If personnel 'p' has to attend simulation event training on day 'd' and scheduled for alert on day d+1, then the personnel must be assigned to any of the first 3 slots of day 'd'	$\sum_{simEvent} (\text{PersonSimEventSchedule}_{p,d,simEvent} * \text{AssignPersonToSimTraining}_{p,d,simEvent}) * \text{PersAlertSchedule}_{p,d+1,'Alert'} \leq \sum_{sim} (\text{AssignPersonToSlot}_{'1',d,p} + \text{AssignPersonToSlot}_{'2',d,p} + \text{AssignPersonToSlot}_{'3',d,p})$	$\forall$p, d	(C57)

Meaning	Constraints		C#
If instructor 'ins' has to attend TR training on day 'd' and scheduled for alert on day d+1, then the instructor must be assigned to any of the first 3 slots of day 'd'	$\text{PersonSimEventSchedule}_{p,d,'TR'} * \text{PersAlertSchedule}_{ins,d+1,'Alert'} \leq \sum_{sim}(\text{AssignInstructorsToSlot}_{'1',sim,d,ins} + \text{AssignInstructorsToSlot}_{'2',sim,d,ins} + \text{AssignInstructorsToSlot}_{'3',sim,d,ins})$	$\forall$ ins, d	(C58)
If evaluator 'evals' has to attend annual evaluation simulation training on day 'd' and scheduled for alert on day d+1, then the evaluator must be assigned to any of the first 3 slots of day 'd'	$\text{PersonSimEventSchedule}_{p,d,'eval'} * \text{PersAlertSchedule}_{evals,d+1,'Alert'} \leq \sum_{sim}(\text{AssignEvaluatorsToSlot}_{'1',sim,d,evals} + \text{AssignEvaluatorsToSlot}_{'2',sim,d,evals} + \text{AssignEvaluatorsToSlot}_{'3',sim,d,evals})$	$\forall$ evals, d	(C59)
To set (save) person simulation event training schedule	$\text{PersonSchedule}_{p,d,simEvent} = \text{PersonSimEventSchedule}_{p,d,simEvent}$	$\forall$ p, d, simEvent	(C60)
To set (save) person training class schedule	$\text{PersonSchedule}_{p,d,class} = \text{PersonClassSchedule}_{p,d,class}$	$\forall$ p, d, class	(C61)
To set (save) backup personnel schedule	$\text{PersonSchedule}_{p,d,'Backup'} = \text{PersBackupSchedule}_{p,d,'Backup'}$	$\forall$ p, d	(C62)
	$\sum_{sim}\text{AssignInstructorsToSlot}_{'1',sim,d,ins} + \sum_{sim}\text{AssignInstructorsToSlot}_{'3',sim,d,ins} + \sum_{sim}\text{AssignInstructorsToSlot}_{'5',sim,d,ins} \leq 1$	$\forall$ ins, d	(C63)
	$\sum_{sim}\text{AssignInstructorsToSlot}_{'2',sim,d,ins} + \sum_{sim}\text{AssignInstructorsToSlot}_{'4',sim,d,ins} \leq 1$	$\forall$ ins, d	(C64)
	$\sum_{sim}\text{AssignInstructorsToSlot}_{'1',sim,d,ins} + \sum_{sim}\text{AssignInstructorsToSlot}_{'4',sim,d,ins} \leq 1$	$\forall$ ins, d	(C65)
	$\sum_{sim}\text{AssignInstructorsToSlot}_{'2',sim,d,ins} + \sum_{sim}\text{AssignInstructorsToSlot}_{'5',sim,d,ins} \leq 1$	$\forall$ ins, d	(C66)
Crew Integrity Constraints			

Meaning	Constraints		C#
To identify if the crew mates are paired for event 'e'	$2 * \text{PersPairedWithMateEvent}_{p,d,e} \leq \text{PersonSchedule}_{p,d,e} + \text{PersonSchedule}_{p2,d,e}$	$\forall$ p, p2, d, e CrewMates(p,p 2)=1	(C67)
To identify if the crew mates are paired at site 'site'	$2 * \text{PersPairedWithMateSite}_{p,d,site} \leq \text{AssignPersToSite}_{p,site,d} + \text{AssignPersToSite}_{p2,site,d}$	$\forall$ p, p2, d, site CrewMates(p,p 2)=1	(C68)
To identify if the crew mates are paired in training class 'class'	$2 * \text{PersPairedWithMateClass}_{p,d,class} \leq \text{AssignPersonToTrainingClass}_{p,d,class} + \text{AssignPersonToTrainingClass}_{p2,d,class}$	$\forall$ p, p2, d, class CrewMates(p,p 2)=1	(C69)
To identify if the crew mates are paired in simulation event training 'simEvent'	$2 * \text{PersPairedWithMateSimEvent}_{p,d,simEvent} \leq \text{AssignPersonToSimTraining}_{p,d,simEvent} + \text{AssignPersonToSimTraining}_{p2,d,simEvent}$	$\forall$ p, p2, d, simEve CrewMates(p,p 2)=1	(C70)
To identify if the crew mates are paired in simulation event training 'simEvent' same slot 'sl' in a simulator	$2 * \text{PersPairedWithMateSimEvent}_{p,d,simEvent} \leq \text{AssignPersonToSlot}_{sl,sim,d,p} + \text{AssignPersonToSlot}_{sl,sim,d,p2}$	$\forall$ p, p2, d, simEvent, sl, sim CrewMates(p,p 2)=1	(C71)
To identify if the crew mates are paired for event 'e'	$2 * \text{PersPairedWithMateBackup}_{p,d} \leq \text{PersonSchedule}_{p,d,'Backup'} + \text{PersonSchedule}_{p2,d,'Backup'}$	$\forall$ p, p2, d CrewMates(p,p 2)=1	(C72)
Backup Crew Constraints			
To assign a deputy to the backup crew	$\sum_{p} \text{PersAssignedAsBackupCrew}_{p,d} * \text{PersToPos}_{p,'Deputy'} + \text{PersAlertAvailDays}_{p,d} = 1$	$\forall$ d	(C73)
To assign a commander to the backup crew	$\sum_{p} \text{PersAssignedAsBackupCrew}_{p,d} * \text{PersToPos}_{p,'Commander'} + \text{PersAlertAvailDays}_{p,d} = 1$	$\forall$ d	(C74)
To confirm that the scheduled backup personnel is not scheduled for an alert	$\text{PersAssignedAsBackupCrew}_{p,d} \leq 1 - (\text{PersonSchedule}_{p,d,'Alert'} + \text{PersonSchedule}_{p,d,'O-day'}) + \sum_{class} \text{PersonSchedule}_{p,d,class} +$ simEventPersonSchedule p, d, simeEvent	$\forall$ p, d	(C75)

Meaning	Constraints		C#
To confirm that the scheduled backup personnel is not scheduled for an O-day the day before	$\text{PersAssignedAsBackupCrew}_{p,d} \leq 1 - \text{PersonSchedule}_{p,d-1,'O-day'}$	$\forall p, d$	(C76)
To confirm that the scheduled backup personnel is not scheduled for a a shift the following 2 days	$\text{PersAssignedAsBackupCrew}_{p,d} \leq 1 - (\text{PersonSchedule}_{p,d+1,'Alert'} + \text{PersonSchedule}_{p,d+2,'Alert'})$	$\forall p, d$	(C77)
To not assign a person to backup on day 'd' if the total number of assigned alerts in the month is as much as the max they're allowed to do for their particular job title	$\text{PersAssignedAsBackupCrew}_{p,d} \leq \sum_{j} (\text{MaxAlertsPerJob}_j * \text{PersToJob}_{p,j}) - \sum_{d2} \text{PersonSchedule}_{p,d2,'Alert'}$	$\forall p, d$	(C78)
To assign the personnel as a backup crew in the schedule.	$\text{PersAssignedAsBackupCrew}_{p,d} = \text{PersonSchedule}_{p,d,'Backup'}$	$\forall p, d$	(C79)
Different Days Penalty Constraints			
	$\sum_{j} (\text{AverageDaysOnAlertJob}_j * \text{PersToJob}_{p,j}) - \text{NumDaysPersonOnAlert}_p \leq \text{DiffDaysOnAlert}_p$	$\forall p$	(C80)
	$-\text{DiffDaysOnAlert}_p \leq \sum_{j} (\text{AverageDaysOnAlertJob}_j * \text{PersToJob}_{p,j}) - \text{NumDaysPersonOnAlert}_p$	$\forall p$	(C81)
	$\text{AverageDaysInsTeachingClass} - \text{NumDaysInsTeachingClass}_{ins} \leq \text{DiffDaysInsTeachingClass}_{ins}$	$\forall ins$	(C82)
	$-\text{DiffDaysInsTeachingClass}_{ins} \leq \text{AverageDaysInsTeachingClass} - \text{NumDaysInsTeachingClass}_{ins}$	$\forall ins$	(C83)
	$\text{AverageDaysInsTeachingTR} - \text{NumDaysInsTeachingTRs}_{ins} \leq \text{DiffDaysInsTeachingTR}_{ins}$	$\forall ins$	(C84)
	$-\text{DiffDaysInsTeachingTR}_{ins} \leq \text{AverageDaysInsTeachingTR} - \text{NumDaysInsTeachingTRs}_{ins}$	$\forall ins$	(C85)

Meaning	Constraints		C#
	$\text{AverageDaysEvalsTeachingEvals} - \text{NumDaysEvalsTeachingEvals}_{evals} \leq \text{DiffDaysEvalsTeachEvals}_{evals}$	$\forall$ evals	(C86)
	$-\text{DiffDaysEvalsTeachEvals}_{evals} \leq \text{AverageDaysEvalsTeachingEvals} - \text{NumDaysEvalsTeachingEvals}_{evals}$	$\forall$ evals	(C87)
	$\sum_j (\text{AverageDaysOnBackup}_j * \text{PersToJob}_{p,j}) - \text{NumDaysPersonOnBackup}_p \leq \text{DiffDaysOnBackup}_p$	$\forall$ p	(C88)
	$-\text{DiffDaysOnBackup}_p \leq \sum_j (\text{AverageDaysOnBackup}_j * \text{PersToJob}_{p,j}) - \text{NumDaysPersonOnBackup}_p$	$\forall$ p	(C89)
Dynamic Portion Constraints			
To save/keep the overall schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{PersonSchedule}_{p,d,e} = \text{CompletedSchedule}_{p,d,e}$	$\forall$ p, d, e ($d \leq P26 - 1$)	(C90)
To save/keep the "alert" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignPersToSite}_{p,site,d} = \text{CompletedAlertAssignments}_{p,site,d}$	$\forall$ p, site, d ($d \leq P26 - 1$)	(C91)
To save/keep the "training classes" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignPersonToTrainingClass}_{p,d,class} = \text{CompletedClassTrainingAssignments}_{p,d,class}$	$\forall$ p, d, class ($d \leq P26 - 1$)	(C92)
To save/keep the "simulation training events" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignPersonToSimTraining}_{p,d,simEvent} = \text{CompletedSimTrainingAssignments}_{p,d,simEvent}$	$\forall$ p, d, simEvent ($d \leq P26 - 1$)	(C93)
To save/keep the instructors "training classes" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignInstructorsToTrainClass}_{ins,d,class} = \text{CompletedInsTrainAssignments}_{ins,d,class}$	$\forall$ ins, d, class ($d \leq P26 - 1$)	(C94)
To save/keep the instructors "TR training" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignInstructorsToTrainTRs}_{ins,d,'TR'} = \text{CompletedInsTrainAssignments}_{ins,d,'TR'}$	$\forall$ ins, d ($d \leq P26 - 1$)	(C95)

Meaning	Constraints		C#
To save/keep the evaluators "annual evaluation training classes" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignEvaluatorsToEvals}_{\text{evals,d,'eval'}} = \text{CompletedEvalTrainAssignments}_{\text{evals,d,'eval'}}$	$\forall$ evals, d $(d \leq P26 - 1)$	(C96)
To save/keep the assigned slots of the "simulator" schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignPersonToSlot}_{\text{sl,sim,d,p}} = \text{CompletedAssignPersonToSlot}_{\text{sl,sim,d,p}}$	$\forall$ sl, sim, d, p $(d \leq P26 - 1)$	(C97)
To save/keep the evaluators simulator slots schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignEvaluatorsToSlot}_{\text{sl,sim,d,'eval'}} = \text{CompletedEvalToSlot}_{\text{sl,sim,d,'eval'}}$	$\forall$ sl, sim, d, evals $(d \leq P26 - 1)$	(C98)
To save/keep the instructors simulator slots schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignInstructorsToSlot}_{\text{sl,sim,d,ins}} = \text{CompletedInsToSlot}_{\text{sl,sim,d,ins}}$	$\forall$ sl, sim, d, ins $(d \leq P26 - 1)$	(C99)
To save/keep the simulator to slots schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{AssignSimulatorToSlot}_{\text{sl,d,sim}} = \text{CompletedSimToSlot}_{\text{sl,d,sim}}$	$\forall$ sl, d, sim $(d \leq P26 - 1)$	(C100)
To save/keep the number of slots per instructor schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{NumberSlotsPerInstructor}_{\text{d,ins}} = \text{CompletedNumSlotsPerIns}_{\text{d,ins}}$	$\forall$ d, ins $(d \leq P26 - 1)$	(C101)
To save/keep the simulator to simulation event training slots schedule up to the day $\mathbf{d \leq P26 - 1}$	$\text{SlotUsedForSimEvent}_{\text{sl,sim,d,simEvent}} = \text{CompletedSlotForSimEvent}_{\text{sl,sim,d,simEvent}}$	$\forall$ sl, sim, d, simEvent $(d \leq P26 - 1)$	(C102)
	$\text{OriginalSchedule}_{\text{p,d,e}} - \text{PersonSchedule}_{\text{p,d,e}} \leq \text{DiffOrigSchedule}_{\text{p,d,e}}$	$\forall$ p, d, e	(C103)
	$-\text{DiffOrigSchedule}_{\text{p,d,e}} \leq \text{OriginalSchedule}_{\text{p,d,e}} - \text{PersonSchedule}_{\text{p,d,e}}$	$\forall$ p, d, e	(C104)
	$\text{OriginalAlertAssignments}_{\text{p,site,d}} - \text{AssignPersToSite}_{\text{p,site,d}} \leq \text{DiffOrigAlertAssignments}_{\text{p,site,d}}$	$\forall$ p, site, d	(C105)
	$-\text{DiffOrigAlertAssignments}_{\text{p,site,d}} \leq \text{OriginalAlertAssignments}_{\text{p,site,d}} - \text{AssignPersToSite}_{\text{p,site,d}}$	$\forall$ p, site, d	(C106)
	$\text{OriginalClassTrainingAssignments}_{\text{p,d,class}} - \text{AssignPersonToTrainingClass}_{\text{p,d,class}} \leq \text{DiffOrigClassTrainingAssignments}_{\text{p,d,class}}$	$\forall$ p, d, class	(C107)

Meaning	Constraints		C#
	$-\text{DiffOrigClassTrainingAssignments}_{p,d,class}$ $\leq \text{OriginalClassTrainingAssignments}_{p,d,class}$ $- \text{AssignPersonToTrainingClass}_{p,d,class}$	$\forall$ p, d, class	(C108)
	$\text{OriginalSimTrainingAssignments}_{p,d,simEvent}$ $- \text{AssignPersonToSimTraining}_{p,d,simEvent}$ $\leq \text{DiffOrigSimTrainingAssignment}_{p,d,simEvent}$	$\forall$ p, d, simEvent	(C109)
	$-\text{DiffOrigSimTrainingAssignment}_{p,d,simEvent}$ $\leq \text{OriginalSimTrainingAssignments}_{p,d,simEvent}$ $- \text{AssignPersonToSimTraining}_{p,d,simEvent}$	$\forall$ p, d, simEvent	(C110)
	$\text{OriginalInstructorTrainAssignments}_{ins,d,class}$ $- \text{AssignInstructorsToTrainClass}_{ins,d,class}$ $\leq \text{DiffOrigInsTrainClassAssignments}_{ins,d,class}$	$\forall$ ins, d, class	(C111)
	$-\text{DiffOrigInsTrainClassAssignments}_{ins,d,class}$ $\leq \text{OriginalInstructorTrainAssignments}_{ins,d,class}$ $- \text{AssignInstructorsToTrainClass}_{ins,d,class}$	$\forall$ ins, d, class	(C112)
	$\text{OriginalInstructorTrainAssignments}_{ins,d,'TR'}$ $- \text{AssignInstructorsToTrainTRs}_{ins,d,'TR'}$ $\leq \text{DiffOrigInsTrainTRAssignments}_{ins,d,'TR'}$	$\forall$ ins, d	(C113)
	$-\text{DiffOrigInsTrainTRAssignments}_{ins,d,'TR'}$ $\leq \text{OriginalInstructorTrainAssignments}_{ins,d,'TR'}$ $- \text{AssignInstructorsToTrainTRs}_{ins,d,'TR'}$	$\forall$ ins, d	(C114)
	$\text{OriginalEvalTrainAssignments}_{evals,d,'eval'} - \text{AssignEvaluatorsToEvals}_{evals,d,'eval'}$ $\leq \text{DiffOrigEvalTrainAssignments}_{evals,d,'eval'}$	$\forall$ evals, d	(C115)
	$-\text{DiffOrigEvalTrainAssignments}_{evals,d,'eval'}$ $\leq \text{OriginalEvalTrainAssignments}_{evals,d,'eval'}$ $- \text{AssignEvaluatorsToEvals}_{evals,d,'eval'}$	$\forall$ evals, d	(C116)
	$\text{OriginalBackupCrew}_{p,d} - \text{PersAssignedAsBackupCrew}_{p,d}$ $\leq \text{DiffOrigBackupCrew}_{p,d}$	$\forall$ p, d	(C117)
	$-\text{DiffOrigBackupCrew}_{p,d}$ $\leq \text{OriginalBackupCrew}_{p,d} - \text{PersAssignedAsBackupCrew}_{p,d}$	$\forall$ p, d	(C118)

Table 8: Output Parameters of the AFOCS model

Parameter	Meaning		O #
$\mathbf{OriginalSchedule_{p,d,e}}$	$= \begin{cases} 1, \text{personnel 'p' is scheduled for event 'e' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V53$	(O1)
$\mathbf{OriginalAlertAssignments_{p,site,e}}$	$= \begin{cases} 1, \text{personnel 'p' is scheduled for site 'site' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V1$	(O2)
$\mathbf{OriginalClassTrainingAssignmentse_{p,d,class}}$	$= \begin{cases} 1, \text{personnel 'p' is scheduled for training 'class' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V2$	(O3)
$\mathbf{OriginalSimTrainingAssignments_{p,d,simEven}}$	$= \begin{cases} 1, \text{personnel 'p' is scheduled for training 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V3$	(O4)
$\mathbf{OriginalInstructorTrainAssignments_{ins,d,tr}}$	$= \begin{cases} 1, \text{instructor 'ins' is scheduled to instruct 'train' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V4\&V5$	(O5)
$\mathbf{OriginalEvalTrainAssignments_{evals,d,simEven}}$	$= \begin{cases} 1, \text{evaluator 'evals' is scheduled to instruct 'simEvent' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V6$	(O6)
$\mathbf{OriginalAssignPersonToSlot_{sl,sim,d,p}}$	$= \begin{cases} 1, \text{slot 'sl' of simulator 'sim' is assigned to personnel 'p' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V10$	(O7)
$\mathbf{OriginalEvalToSlot_{sl,sim,d,evals}}$	$= \begin{cases} 1, \text{evaluator 'evals' is assigned to slot 'sl' of simulator 'sim' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V11$	(O8)
$\mathbf{OriginalInsToSlot_{sl,sim,d,ins}}$	$= \begin{cases} 1, \text{instructor 'ins' is assigned to slot 'sl' of simulator 'sim' on day 'd'} \\ 0, \text{else} \end{cases}$	$= V12$	(O9)
$\mathbf{OriginalSimToSlot_{sl,d,sim}}$	$= \begin{cases} 1, \text{slot 'sl' of simulator 'sim' is assigned on day 'd'} \\ 0, \text{else} \end{cases}$	$= V13$	(O10)
$\mathbf{OriginalNumSlotsPerIns_{d,in}}$	$=$ number of slots per instructor on day 'd'	$= V14$	(O11)
$\mathbf{OriginalSlotForSimEvent_{sl,sim,d,simEvent}}$	$= \begin{cases} 1, \text{slot 'sl'of simulator 'sim' is assigned on day 'd'forsimluation'simEvent'} \\ 0, \text{else} \end{cases}$	$= V15$	(O12)
$\mathbf{OriginalBackupCrew_{p,d}}$	$= \begin{cases} 1, \text{personnel 'p' is scheduled as backup on day 'd'} \\ 0, \text{else} \end{cases}$	$= V28$	(O13)

12.2.5 Model Execution

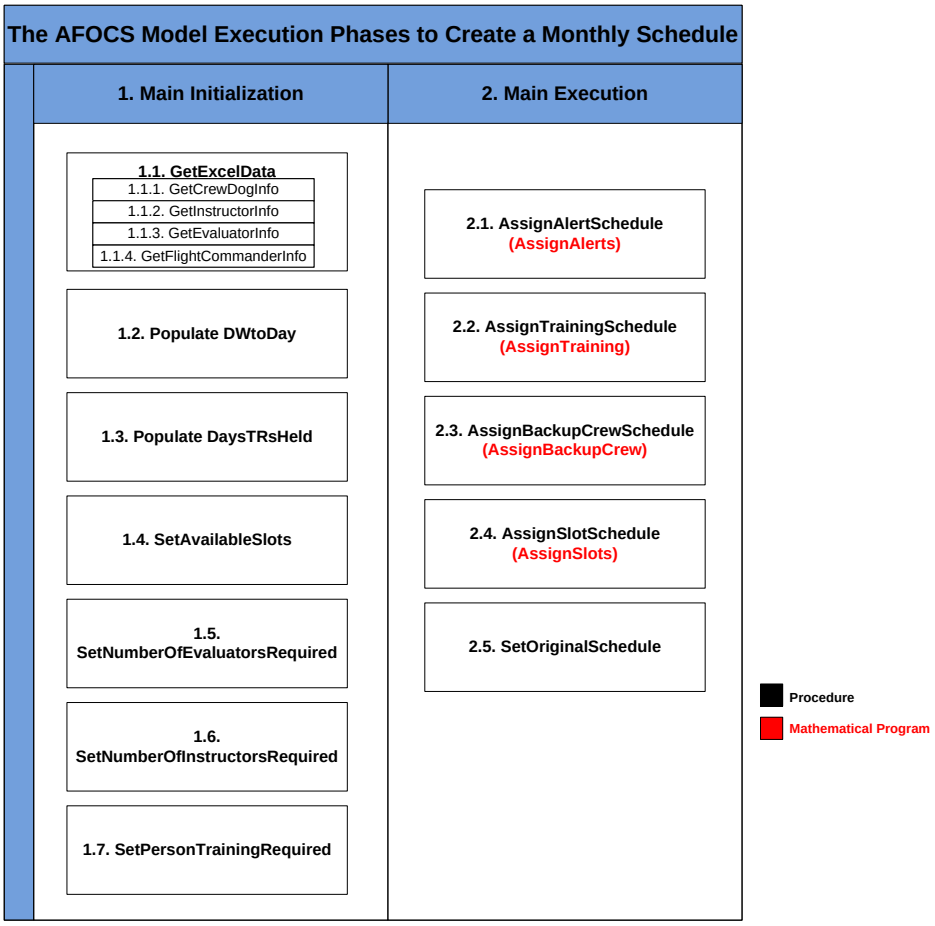
The model is can be used for two purposes, to:

1- Create a Monthly Schedule
2- Re-Create a Monthly Schedule (re-create an existing schedule base on unexpected changes, such as personnel availability.)

12.2.5.1 Create a Monthly Schedule

The model potion to create a monthly schedule contains many procedures and mathematical programs. Figure 7 shows the breakdown of this potion.

Figure 7: The AFOCS Model Execution Phases to Create a Monthly Schedule



12.2.5.1.1 Main Initialization

This phase is broken down into 7 procedures, and the main objective of this phase is to prepare the model to execute and solve the mathematical programs.

Before triggering the first procedure "GetExcelData", there are two command lines:

```
empty AlertReq;
AlertReq(site,d,pos) := 1;
GetExcelData;
PopulateDWtoDay;
PopulateDaysTRsHeld;
SetAvailableSlots;
SetNumberOfEvaluatorsRequired;
SetNumberOfInstructorsRequired;
SetPersonTrainingRequired;
```

The first command is to empty parameter P20 from any previous values (previous month results, or previous model execution). And the second command is to set parameter P20 to 1, which means all sites, for every day of the month, and every position (commander and deputy) is required to be filed (scheduling the alerts).

GetExcelData

The main action of this procedure is to extract data from a database (excel spreadsheet "*AFOCSInputFile.xlsx"*) and assign them to some parameters, which are personnel info such as, job, position, availability, squadron, site, and other.

This procedure is broken into four sub-procedures GetCrewDogInfo, GetInstructorInfo, GetEvaluatorInfo, GetFlightCommanderinfo. All those four sub-procedures extract the same data for each job role, which are Crew Members, Instructors, Evaluators and Flight Commanders.

Parameter P7-P16 will be defined by the end of this procedure.

```
empty Personnel;
empty PersToJob;
empty PersAlertAvailDays;
empty PersTrainAvailDays;
empty PersToPos;
empty PersToJob;
empty PersSCPCertified;
empty PersToSquadron;

GetCrewDogInfo;
GetInstructorInfo;
GetEvaluatorInfo;
GetFlightCommanderInfo;
for(p in personnel, d in day) do
        if(PersAlertAvailDays(p,d) = 1) then
                PersAlertAvailDays(p,d) := 0;
        else PersAlertAvailDays(p,d) := 1;
        endif;
        if(PersTrainAvailDays(p,d) = 1) then
                PersTrainAvailDays(p,d) := 0;
        else PersTrainAvailDays(p,d) := 1;
        endif
```

```
endfor;
ExcelCloseWorkbook(InputWorkbookName,1);
ExcelSetVisibility(InputWorkbookName, 'On');
```

Populate DWtoDay

Based on the given parameters P1, P2, P4 and P5, this procedure populates parameter P3, which is assigning the day of the week (Su, M, T, W, Th, F, S) to the day of the month (1,2,…,end of the month)

```
empty DWtoDay;
for (dw in dayofweek) do
        if (FirstDayOfMonthDW(InputMonth,InputYear,dw) = 1) then
                DWtoDay('1',dw) := 1;
                DWtoDay('8',dw) := 1;
                DWtoDay('15',dw) := 1;
                DWtoDay('22',dw) := 1;
                DWtoDay('29',dw) := 1;
        endif
endfor;

for(dw in dayOfweek, d in day) do
        if(DWtoDay(d,dw) = 1) then
                DWtoDay(d+1,dw+1) := 1;
        endif;
        if(DWtoDay(d,'Su') = 1) then
                DWtoDay(d+1,'M') := 1;
                DWtoDay(d+2,'T') := 1;
                DWtoDay(d+3,'W') := 1;
                DWtoDay(d+4,'Th'):= 1;
                DWtoDay(d+5,'F') := 1;
                DWtoDay(d+6,'S') := 1;
        endif;
endfor;

for (d in day | Val(d) > NumDaysInMonth(InputMonth, InputYear)) do
        DWtoDay(d,dw) := 0;
endfor;
```

Populate DaysTRsHeld

This procedure is to populate parameter P24 which is the days that TR training is available (Monday thru Saturday).

```
for (d in Day) do
        if(DWtoDay(d,'Su') = 0) then
                DaysSimsAvailable(d) := 1;
        else
                DaysSimsAvailable(d) := 0;
        endif
endfor
```

SetAvailableSlots

Based on the given parameters P22 and P24, this procedure sets the number of available slots (4 or 5) per day, if the simulator is available at that particular day. This procedure sets parameters P27 and P28.

```
for(d in day) do
        if (DaysSimsAvailable(d) = 1) then
                if(NumberOfSlotsForTraining = 4) then
                        SlotsAvailable('1',d) := 1;
                        SlotsAvailable('2',d) := 1;
                        SlotsAvailable('3',d) := 1;
                        SlotsAvailable('4',d) := 1;
                elseif (NumberOfSlotsForTraining = 5) then
                        SlotsAvailable('1',d) := 1;
                        SlotsAvailable('2',d) := 1;
                        SlotsAvailable('3',d) := 1;
                        SlotsAvailable('4',d) := 1;
                        SlotsAvailable('5',d) := 1;
                endif
        else
                SlotsAvailable(sl,d) := 0;
        endif;
        SimsAvailable(sim,d) := DaysSimsAvailable(d);
endfor;
```

SetNumberOfInstructorsRequired

This is to set the number of instructors required for the training events (TR, T1, T3, T4). From the requirements the number of instructors must be 2, so P25 =2.

```
NumberOfInstructorsRequired(d,train) := 2;
```

SetNumberofEvaluatorsRequired

This is to set the number of evaluators required for the Annual Review “evals” training event. From the requirements the number of evaluators must be 3, so P26 =3.

```
NumberOfEvaluatorsRequired(d,'eval') := 3;
```

SetPersonTrainingRequired

From the requirements all personnel must attend monthly training and annual review training if they are scheduled for it. This procedure sets parameters P17 and P18.

```
PersClassTrainingReq(p,'T1') := 1;
PersClassTrainingReq(p,'T3') := 1;
PersClassTrainingReq(p,'T4') := 1;
PersClassTrainingReq(p, 'T3/T4') := 0;
PersSimEventReq(p, 'TR') := 1;

PersSimEventReq('Person1', 'Eval') := 1;
PersSimEventReq('Person2', 'Eval') := 1;
PersSimEventReq('Person81', 'Eval') := 1;
PersSimEventReq('Person82', 'Eval') := 1;
```

12.2.5.1.2 Main Execution

After "Main Initialization" procedure executes and from the "user" input data, parameters P1-P28 are defined, and the "Main Execution" procedure takes place. This procedure is broken down into four sub-procedures.

```
AssignAlertSchedule;
AssignTrainingSchedule;
AssignSlotSchedule;
SetOriginalSchedule;
```

AssignAlertSchedule

The outcome of this procedure are the shifts and the O-day schedule for all personnel throughout the whole month.

```
empty AssignPersToSite;
empty PersonSchedule;
empty NumDaysPersonOnAlert;
empty ObjectiveFunctionAlert;
empty PersAlertSchedule;
solve AssignAlerts;
PersAlertSchedule(p,d,'Alert') := PersonSchedule(p,d,'Alert');
PersAlertSchedule(p,d,'O-day') := PersonSchedule(p,d,'O-day');
empty PersonSchedule(p,d,'B-day');
```

The mathematical program "AssignAlerts" executes in this procedure:

```
Objective function: Maximize V57
Constraints: AlertConstraints
Variables: AlertVariables
```

After the objective function (mathematical program) is solved, P46 is being populated with shifts and the O-day schedule.

AssignTrainingSchedule

The outcome of this procedure is all training events schedule for all personnel throughout the whole month.

```
empty PersonSchedule(p,d,train);
empty PersonSchedule(p,d,simEvent);
empty AssignPersonToTrainingClass;
empty AssignPersonToSimTraining;
empty AssignInstructorsToTrainClass;
empty AssignInstructorsToTrainTRs;
empty AssignEvaluatorsToEvals;
empty ClassroomTrainingCompleted;
empty ClassRequirement;
empty ObjectiveFunctionTraining;
empty PersonSimEventSchedule;
empty PersonClassSchedule;
solve AssignTraining;
PersonSimEventSchedule(p,d,'TR') := PersonSchedule(p,d,'TR');
PersonSimEventSchedule(p,d,'Eval') := PersonSchedule(p,d,'Eval');
```

```
PersonClassSchedule(p,d,'T1') := PersonSchedule(p,d,'T1');
PersonClassSchedule(p,d,'T3') := PersonSchedule(p,d,'T3');
PersonClassSchedule(p,d,'T4') := PersonSchedule(p,d,'T4');
PersonClassSchedule(p,d,'T3/T4') := PersonSchedule(p,d,'T3/T4');
```

The mathematical program "AssignTraining" executes in this procedure:

```
Objective function: Maximize V58
Constraints: TrainingConstraints
Variables: TrainingVariables
```

After the objective function (mathematical program) is solved, P47 and P48 are being populated with training events schedule, which includes the schedule for all training classes and simulators.

AssignBackupCrewSchedule
The outcome of this procedure is the backup crew schedule throughout the whole month. The mathematical program "AssignBackupCrew" executes in this procedure:

```
Objective function: Minimize V58
Constraints: BackupCrewConstraints
Variables: BackupCrewVariables
```

After the objective function (mathematical program) is solved, P49 is being populated with the backup crew schedule.

AssignSlotSchedule
The outcome of this procedure is all simulator training slots schedule for all personnel throughout the whole month.

```
empty AssignPersonToSlot;
empty AssignEvaluatorsToSlot;
empty AssignInstructorsToSlot;
empty NumberSlotsPerInstructor;
empty SlotUsedForSimEvent;
empty AssignSimulatorToSlot;
solve AssignSlots;
```

The mathematical program "AssignSlots" executes in this procedure:

```
Objective function: Maximize V58
Constraints: SlotConstraints
Variables: SlotVariables
```

After the objective function (mathematical program) is solved, slots are assigned to different personnel (crew members, instructors, evaluators and flight commanders).

SetOriginalSchedule
This is the last step of the overall execution, the outcome of this procedures it the monthly schedule for all the shifts and the training events. The procedure it assigns the values for all the output parameters O1-O13.

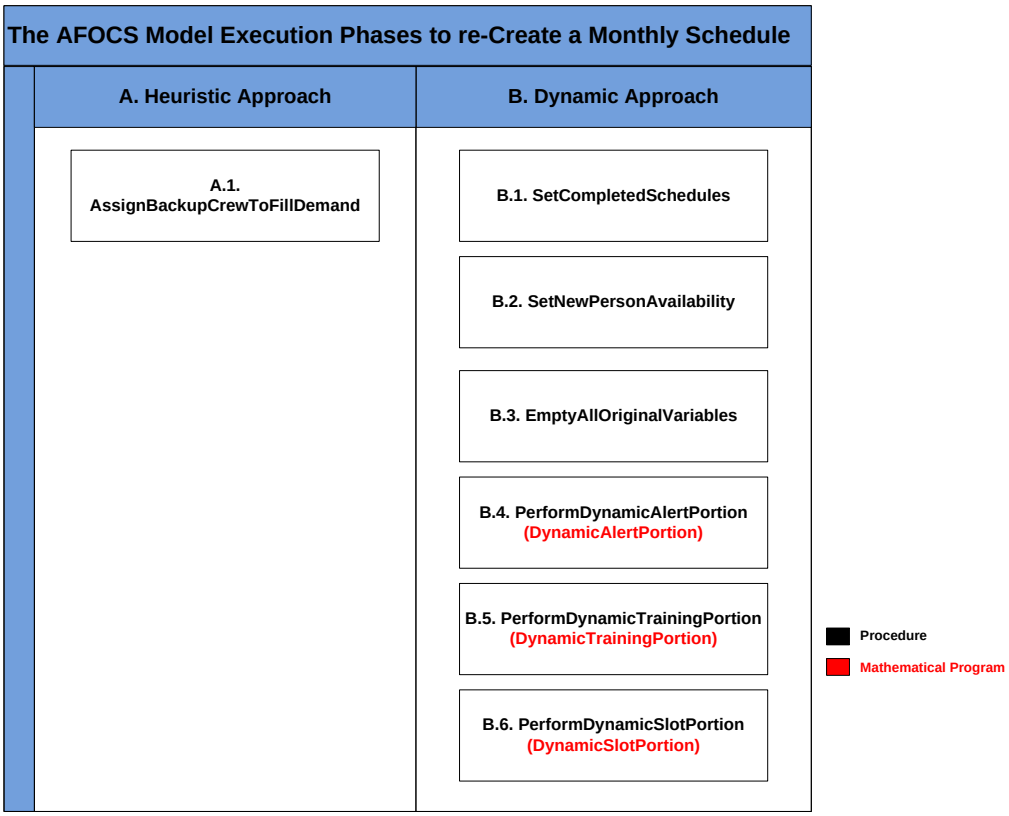
```
OriginalSchedule(p,d,e)                    := PersonSchedule(p,d,e);
OriginalAlertAssignments(p,site,d)         := AssignPersToSite(p,site,d);
OriginalEvalTrainAssignments(evals,d,'eval')   := AssignEvaluatorsToEvals(evals,d,'eval');
OriginalInstructorTrainAssignments(ins,d,'TR') := AssignInstructorsToTrainTRs(ins,d,'TR');
OriginalBackupCrew(p,d)                    := PersAssignedAsBackupCrew(p,d);
OriginalAssignPersonToSlot(sl,sim,d,p)     := AssignPersonToSlot(sl,sim,d,p);
OriginalEvalToSlot(sl,sim,d,evals)         := AssignEvaluatorsToSlot(sl,sim,d,evals);
OriginalInsToSlot(sl,sim,d,ins)            := AssignInstructorsToSlot(sl,sim,d,ins);
OriginalSimToSlot(sl,d,sim)                := AssignSimulatorToSlot(sl,d,sim);
OriginalNumSlotsPerIns(d,ins)              := NumberSlotsPerInstructor(d,ins);
OriginalSlotForSimEvent(sl,sim,d,simEvent) := SlotUsedForSimEvent(sl,sim,d,simEvent);

for (e in events) do
        if (e in classroomEvents) then
                for (class in classroomEvents) do
                OriginalClassTrainingAssignments(p,d,class) :=
AssignPersonToTrainingClass(p,d,class);
                OriginalInstructorTrainAssignments(ins,d,class) :=
AssignInstructorsToTrainClass(ins,d,class);
                endfor
        elseif (e in simulationEvents) then
                for (simEvent in simulationEvents) do
                OriginalSimTrainingAssignments(p,d,simEvent) :=
AssignPersonToSimTraining(p,d,simEvent);
                endfor
        endif
endfor
```

12.2.5.2 Re-Create a Monthly Schedule

The previous section covers the execution to create a monthly schedule. This section covers when personnel availability changes during the month. Two methods were developed to address those daily changes. The model potion to re-create a monthly schedule contains many procedures and mathematical programs. Figure 8 shows the breakdown of this potion.

Figure 8: The AFOCS Model Execution Phases to re-Create a Monthly Schedule



12.2.5.2.1 Heuristic Approach

This is for situations where 1-2 people are unavailable on a particular day (ProblemDay) for alerts. There is a backup crew scheduled each day (from the previous section – Create a Monthly Schedule) for these situations. If a person becomes unavailable, then the backup crew is pulled to replace them. So if a Deputy becomes unavailable, the Deputy on the backup crew gets pulled for alert.

AssignBackupCrewToFillDemand

```
SetNewPersonAvailability;
for (p in Personnel | PersAssignedAsBackupCrew(p,ProblemDay) = 1) do
        ! If the person in the backup crew is available and has the same position as the
PersonNowUnavailable, choose them
        if (
                (
                PersToPos(p,'Commander') = PersToPos(PersonNowUnavailable,'Commander')
                OR PersToPos(p,'Deputy') = PersToPos(PersonNowUnavailable,'Deputy')
                )
                AND PersAlertAvailDays(p,ProblemDay) = 1
                )
                then
                for (site in sites) do
```

```
                    ! Assign the backup person to the site that PersonNowUnavailable was
assigned to
                    AssignPersToSite(p,site,ProblemDay) :=
AssignPersToSite(PersonNowUnavailable,site,ProblemDay);
                    AssignPersToSite(PersonNowUnavailable,site,ProblemDay) := 0;
                    ! Change the person's schedule
                    PersonSchedule(PersonNowUnavailable,ProblemDay,'Alert') := 0;
                    PersonSchedule(PersonNowUnavailable,ProblemDay + 1,'O-day') := 0;
                    PersAlertSchedule(PersonNowUnavailable,ProblemDay,'Alert') := 0;
                    PersAlertSchedule(PersonNowUnavailable,ProblemDay + 1,'O-day') := 0;
                    ! change the backup person's schedule
                    PersonSchedule(p,ProblemDay,'Alert') := 1;
                    PersonSchedule(p,ProblemDay + 1,'O-day') := 1;
                    PersonSchedule(p,ProblemDay,'Backup') := 0;
                    PersAssignedAsBackupCrew(p,ProblemDay) := 0;
                    PersAlertSchedule(p,ProblemDay,'Alert') := 1;
                    PersAlertSchedule(p,ProblemDay+1,'O-day') := 1;
                endfor;

        endif;
endfor;
! Need to check that the other person is still kept for the below procedure
AssignBackupCrewSchedule;
```

The procedure changes the PersonNowUnavailable and the backup crew member's schedules. Then it replaces that backup person on the backup crew with a new person. That's why the AssignBackupCrew procedure is run again. There's nothing to do after that. The original schedule is used with just that one modification on that day. Now another personnel becomes unavailable on that same day (ProblemDay), then the new backup personnel will fill the slot without having to disrupt the schedule. If a third person becomes unavailable, the “Dynamic Approach” takes place.

12.2.5.2.2 Dynamic Approach

This approach is applicable to situations where multiple people (>2) are unavailable on a particular day (ProblemDay) for alerts. In general, those set of procedures reruns the AssignAlert, AssignTraining, AssignSlots procedures but changes the objective function to minimize the different between the new schedule being set and the OriginalSchedule. The SetOriginalSchedule procedure backs up the initial original schedule created. Then running the DynamicAlert/Training/Slots procedures, those procedures use the same constraints/variables as the original set of procedures but also have the added constraints/variables that set the difference between the two schedules (original and the new one being created after the un-availabilities have been input). The objective functions are to minimize the difference between the two schedules (minimize the disruption in the original schedule) while still meeting the requirements/feasibility of the original schedule.

SetCompletedSchedules
This procedure backs up the original schedule of the month that was created earlier.

```
empty CompletedSchedule;
empty CompletedAlertAssignments;
empty CompletedClassTrainingAssignments;
empty CompletedSimTrainingAssignments;
empty CompletedInsTrainAssignments;
empty CompletedEvalTrainAssignments;

for (d in day | d <= ProblemDay - 1) do
        CompletedSchedule(p,d,e)                        :=      OriginalSchedule(p,d,e);
        CompletedAlertAssignments(p,site,d)             :=
        OriginalAlertAssignments(p,site,d);
        CompletedClassTrainingAssignments(p,d,class)    :=
        OriginalClassTrainingAssignments(p,d,class);
        CompletedSimTrainingAssignments(p,d,simEvent)           :=
        OriginalSimTrainingAssignments(p,d,simEvent);
        CompletedInsTrainAssignments(ins,d,train)       :=
        OriginalInstructorTrainAssignments(ins,d,train);
        CompletedEvalTrainAssignments(evals,d,simEvent) :=
        OriginalEvalTrainAssignments(evals,d,simEvent);
        CompletedAssignPersonToSlot(sl,sim,d,p)         :=
        OriginalAssignPersonToSlot(sl,sim,d,p);
        CompletedEvalToSlot(sl,sim,d,evals)             :=
        OriginalEvalToSlot(sl,sim,d,evals);
        CompletedInsToSlot(sl,sim,d,ins)        :=      OriginalInsToSlot(sl,sim,d,ins);
        CompletedSimToSlot(sl,d,sim)                    :=      OriginalSimToSlot(sl,d,sim);
        CompletedNumSlotsPerIns(d,ins)                  :=      OriginalNumSlotsPerIns(d,ins);
        CompletedSlotForSimEvent(sl,sim,d,simEvent)     :=
        OriginalSlotForSimEvent(sl,sim,d,simEvent);
endfor
```

SetNewPersonAvailability

This procedure sets the personnel new availably.

```
NewPersonAlertAvailability(p,d) := PersAlertAvailDays(p,d);
NewPersonTrainAvailability(p,d) := PersTrainAvailDays(p,d);
!set person unavailabilities
NewPersonAlertAvailability(PersonNowUnavailable, ProblemDay) := 0;
!generate new PersAvailDays(p,d)
PersAlertAvailDays(p,d) := NewPersonAlertAvailability(p,d);
```

EmptyAllOriginalVariables

This procedure clears all the variables in the model, so the model can execute with the new inputs (personnel availability)

```
!Global Variables
empty PersonSchedule;
empty IntegralCrewRate;
empty TotalDiffDaysAlert;
empty TotalDiffDaysTraining;
empty ObjectiveFunctionAlert;

!Alert Variables and Backups
empty AssignPersToSite;
```

```
empty PersAlertSchedule;

empty DiffOrigSchedule;
empty DiffOrigAlertAssignments;

!Training Variables and Backups
empty AssignPersonToTrainingClass;
empty AssignPersonToSimTraining;
empty AssignInstructorsToTrainClass;
empty AssignInstructorsToTrainTRs;
empty AssignEvaluatorsToEvals;
empty ClassroomTrainingCompleted;
empty ClassRequirement;
empty PeopleAssignedToTraining;
empty PersonSimEventSchedule;
empty PersonClassSchedule;

!Slot Variables
empty AssignPersonToSlot;
empty AssignEvaluatorsToSlot;
empty AssignInstructorsToSlot;
empty AssignSimulatorToSlot;
empty NumberSlotsPerInstructor;
empty SlotUsedForSimEvent;
empty SlotsAssigned;

!Crew Integrity Variables
empty PersPairedWithMateEvent;
empty PersPairedWithMateSite;
empty PersPairedWithMateClass;
empty PersPairedWithMateSimEvent;
empty PersonCrewIntegrityAlert;
empty PersonCrewIntegrityEvent;
empty PersonCrewIntegrityClass;
empty PersonCrewIntegritySimEvent;
empty PersonCrewIntegrityAllEvents;

!Backup Crew
empty PersAssignedAsBackupCrew;

!Diff Days Penalty Functions
empty NumDaysPersonOnAlert;
empty NumDaysInsTeachingClass;
empty NumDaysInsTeachingTRs;
empty NumDaysEvalsTeachingEvals;
empty AverageDaysOnAlertJob;
empty AverageDaysInsTeachingClass;
empty AverageDaysInsTeachingTR;
empty AverageDaysEvalsTeachingEvals;
empty DiffDaysOnAlert;
empty DiffDaysInsTeachingClass;
empty DiffDaysInsTeachingTR;
empty DiffDaysEvalsTeachEvals;
```

PerformDynamicAlertPortion

The outcome of this procedure are the shifts and O-day schedule throughout the rest of the month (d >= ProblemDay)

```
!SetCompletedSchedules;
!SetNewPersonAvailability;


solve DynamicAlertPortion;
PersAlertSchedule(p,d,'Alert') := PersonSchedule(p,d,'Alert');
PersAlertSchedule(p,d,'O-day') := PersonSchedule(p,d,'O-day');

for (d in day | d >= ProblemDay) do
        empty PersonSchedule(p,d,class);
        empty PersonSchedule(p,d,simEvent);
        empty PersonSchedule(p,d,'Leave');
endfor;
```

The mathematical program "DynamicAlertPortion" executes in this procedure:

```
Objective function: Minimize V52
Constraints: DynamicAlertConstraints
Variables: DynamicAlertVariables
```

PerformDynamicTrainingPortion

The outcome of this procedure is all training events schedule throughout the rest of the month (d >= ProblemDay)

```
empty PersonSimEventSchedule;
empty PersonClassSchedule;

solve DynamicTrainingPortion;
PersonSimEventSchedule(p,d, 'TR') := PersonSchedule(p,d, 'TR');
PersonSimEventSchedule(p,d, 'Eval') := PersonSchedule(p,d, 'Eval');
PersonClassSchedule(p,d, 'T1') := PersonSchedule(p,d, 'T1');
PersonClassSchedule(p,d, 'T3') := PersonSchedule(p,d, 'T3');
PersonClassSchedule(p,d, 'T4') := PersonSchedule(p,d, 'T4');
PersonClassSchedule(p,d, 'T3/T4') := PersonSchedule(p,d, 'T3/T4');
```

The mathematical program "DynamicTrainingPortion" executes in this procedure:

```
Objective function: Minimize V52
Constraints: DynamicTrainingConstraints
Variables: DynamicTrainingVariables
```

PerformDynamicSlotPortion

The outcome of this procedure is all simulator training slots schedule for all personnel throughout the rest of the month (d >= ProblemDay)

```
empty AssignPersonToSlot;
empty AssignEvaluatorsToSlot;
empty AssignInstructorsToSlot;
```

```
empty NumberSlotsPerInstructor;
empty SlotUsedForSimEvent;
empty AssignSimulatorToSlot;
solve DynamicSlotPortion;
```

The mathematical program "DynamicSlotPortion" executes in this procedure:

```
Objective function: Minimize V52
Constraints: DynamicSlotConstraints
Variables: DynamicSlotVariables
```

12.3 Requirements Verification Matrix

Verification methods: inspection, demonstration, test

Number	Requirement	Method	Comments
5.2.1.1	The model shall schedule 15 op centers plus one standby crew.	Test	Pass 4/30/2011
5.2.1.2	The model shall schedule each op center to be staffed by one Crew Commander and one Deputy Crew Commander.	Test	Pass 4/30/2011
5.2.1.3	The model shall schedule 24 hour shifts (7 a.m. to 7 a.m.) for each op center.	Test	Pass 4/30/2011
5.2.1.4	The model shall schedule an off day (O Day) after each alert.	Test	Pass 4/30/2011
5.2.1.5	The model shall schedule the Crew Commander positions with qualified crew members.	Test	Pass 4/30/2011
5.2.1.6	The model shall schedule the Deputy Crew Commander positions with qualified crew members. Personnel qualified as Crew Commander are able to staff the Deputy Crew Commander position.	Test	Pass 4/30/2011
5.2.1.7	The model shall verify that each crew member have completed required mandatory training in the previous calendar month. See section 6.2.2 Functional Requirements (Mandatory Training).	Test	Pass 4/30/2011
5.2.1.8	Each squadron has one Squadron Command Post (SCP). The model shall schedule the staffing of the SCP by SCP qualified crew members. SCP qualification is an additional qualification beyond the Crew Commander/Deputy Crew Commander position.	Test	Pass 4/30/2011
5.2.1.9	The model shall verify that SCP qualified crew members accomplish one SCP alert at a minimum of every 60 days to maintain certification.	Test	Pass 4/30/2011
5.2.1.10	The model shall schedule each crew member at minimum of one alert every 45 days to maintain certification.	Test	Pass 4/30/2011
5.2.1.11	The model shall schedule Instructors, Evaluators, and Flight Commanders for at most two alerts per calendar month.	Test	Pass 4/30/2011
5.2.1.12	The model shall schedule Crew Members for no more than eight alerts per calendar month.	Test	Pass 4/30/2011

Number	Requirement	Method	Comments
5.2.1.13	Pairing the same CDR with the same DEP is referred to as "crew integrity"; one of the organization's goals is to maximize crew integrity. The model shall schedule crew members paired together 80% of the time. In other words, 80% of the time each person is assigned to a shift or training they are paired with the same partner.	Test	Pass 4/30/2011
5.2.2.1	The model shall schedule monthly and annual required training. Monthly mandatory training consists of TR, T1, T3 and T4. Annual required training consists of a four-hour Annual Evaluation in the simulator. (see Table 1)	Test	Pass 4/30/2011
5.2.2.2	The model shall schedule one or two instructors for each monthly training event.	Test	Pass 4/30/2011
5.2.2.3	The model shall schedule three evaluators for each annual evaluation.	Test	Pass 4/30/2011
5.2.3.1	The model shall be able to rebuild the schedule anytime during the calendar month due to an unforeseen absence of an individual.	Test	Pass 4/30/2011
5.2.3.2	While rebuilding the schedule, the model shall minimize rescheduling of future events due to an unforeseen absence of an individual.	Test	Pass 4/30/2011
5.2.4.1	The model shall be able to run on a server.	Demonstration	Deferred
5.2.4.2	The model that runs on a server shall be accessible from a client workstation.	Demonstration	Deferred
5.2.4.3	The model shall provide an output in a format that can be input to TimePiece.	Demonstration	Deferred
5.2.4.4	The model shall be able to import and export data in XML, CSV, and XLSX formats.	Demonstration	Import: Pass 4/30/2011 Export: Deferred
5.2.5.1	Team AFOCS shall develop an optimization algorithm/model.	Inspection	Done 4/30/2011
5.2.5.2	Team AFOCS shall compare performance of different algorithms/approaches.	Inspection	Deferred
5.2.5.3	Team AFOCS shall develop a Requirements Document.	Inspection	Done 4/30/2011