Operational Squadron Scheduling

THESIS

Khaled S. Almannaei, Major, RBAF
AFIT-ENS-MS-18-S-029

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-18-S-029

OPERATIONAL SQUADRON SCHEDULING

THESIS

Presented to the Faculty
Department of Operational Sciences
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

Khaled S. Almannaei, B.S.
Major, RBAF

22 September 2018

AFIT-ENS-MS-18-S-029

OPERATIONAL SQUADRON SCHEDULING

THESIS

Khaled S. Almannaei, B.S.
Major, RBAF

Committee Membership:

Lt Col A. J. Geyer, Ph.D.
Chair

Dr. R. F. Deckro
Member

AFIT-ENS-MS-18-S-029

Abstract

The 28th Operational Weather Squadron (28th OWS) is responsible for producing and disseminating mission planning and execution weather analyses and forecasts. The squadron must prepare schedules that meet the needs of their mission while dealing with real-world constraints such as time windows, task priorities, and intermittent recurring missions. The 28th OWS's manning consists of active duty, deployed in-place, reserve, civilian and contract personnel. In this research, a scheduling model and algorithm are provided as an approach to crew scheduling for the 28th Operational Weather Squadron. Scheduling in the 28th OWS is complex and can be time consuming. This model will reduce the time and burden of scheduling the squadron.

AFIT-ENS-MS-18-S-029

To Allah, with Whom all things are possible.

To my country

To my parents

To my wife

To my kids

for their unwavering support.

Acknowledgements

I would like to express my sincere appreciation to my advisor Lt Col Andrew Geyer, Ph.D., who has supported me throughout my thesis with his patience. I appreciate his vast knowledge and skill in many areas. Without him, I would not have been able to complete this thesis effort.

I would also like to extend my gratitude to Dr. Richard Deckro for his outstanding support and motivation throughout this process.

Khaled S. Almannaei

Table of Contents

List of Figures

List of Tables

OPERATIONAL SQUADRON SCHEDULING

I. Introduction

1.1 Background

The 28th Operational Weather Squadron (28th OWS) is the United States Central Commandand International Security Assistance Force designated Joint Meteorology and Oceanographic Forecast Unit. It provides weather products directly to U.S. Army, Navy, Marine, Air Force, and Coalition warfighters in support of their operational planning and mission execution across 25 countries spanning 6.43 million square miles on two continents [1]. These mission requirements mean that the 28th OWS must perform multiple distinct and highly technical missions simultaneously 24 hours a day, 7 days per week. With squadron personnel ranging in experience and responsibility as well as the need to send personnel on leave or to outside training, scheduling the right mix of personnel for every shift over a monthly planning period can be a difficult and time-consuming task.

1.2 Motivation

At present, it takes an inordinate number of man-hours each month to generate a minimally suitable personnel schedule. When personnel unexpectedly become unavailable due to illness or outside requirements, it can take additional more man-hours to find a new, minimally disruptive personnel schedule. This can become a major draw on manpower during periods of high operations tempo. In addition, when the individual assigned to scheduling duty is unavailable or reassigned, the process becomes more inefficient and mistake prone, causing a further drain on productivity. A high operations tempo unit like the 28th OWS needs to be able to allocate personnel quickly and effectively to ensure their ability to meet their mission requirements with minimal error or wasted effort.

1.3 Problem Statement

Scheduling has extensive application areas in the world. Scheduling is the process by which the job with different requirements is assigned to resources that complete the job. Scheduling is the method of controlling, optimizing and arranging job and workloads in a production process or manufacturing process. Scheduling is used to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials [15].

The scheduling activity is generally carried out by a scheduler. A scheduler may aim to achieve one or more of many goals, such as, minimizing the completion time, minimizing latency, minimizing the number of tardy jobs, maximizing production, and minimizing penalty. The scheduler must ensure that all requirements, restrictions and deadlines can be met by the process. In general, the scheduler should possess enough knowledge of unit regulations, job requirements, qualifications levels, and similar data. For these reasons, the scheduler is often chosen from the most experienced personnel to perform the scheduling process. While some unit scheduled tasks are manually performed, some unit's scheduled tasks are performed by supporting tools to help the scheduler [13].

This thesis is in response to the 28th OWS request for support. The 28th OWS would like to create a tool to generate feasible and flexible duty schedules accounting for all constraints using tools available on the standard Air Force desktop. Creating the monthly schedule for the 28th OWS requires matching available skill sets with position requirements all while accounting for flight integrity, leave, training, temporary duty elsewhere (TDYs), deployments, days off, and so forth. Usually one or more SNCOs spend hours every month generating the duty schedule.

1.4 Scope of the Research

In this research, a Binary Integer Linear Program (BILP) is formulated whose optimal solution is the best possible 28th OWS schedule for the user defined time horizon based on the unit's stated requirements. Regulation related to flights, jobs, and personnel are taken in consideration and used in the application. Even though the scheduling model is developed for the 28th OWS, it has common features that make it suitable for any organization with similar requirements to operate distinct positions around the clock with personnel with varied levels of experience and training.

1.5 Summary

This chapter provided an overview of the 28th OWS mission and its need for a reliable method to quickly generate optimal personnel schedules. In Chapter 2, previous research into scheduling problems is discussed. Chapter 3 explains the methodology behind and formulation of BILP created to generate optimal 28th OWS schedules. In Chapter 4, the BILP model performance is examined and evaluated under various realistic scenarios. Finally, Chapter 5 provides a final summary of the research, the research conclusions and recommendations for future research.

II. Literature Review

2.1 Chapter Overview

This chapter provides an overview of previous research topics related to crew scheduling theory. The purpose is to introduce the concepts of scheduling and some solution methods which will help to implement similar schedules at the 28th OWS.

2.2 Scheduling Theory

The scheduling problem has attracted much interest from both academia and the operational world [9]. Scheduling concerns the allocation of limited resources to tasks over time. It is a decision-making process that has the goal of optimizing one or more objectives [13]. It exists in almost all operational environments. The problems that scheduling theory are concerned with are described as optimizing a limited set of assignment decisions with limited resources [14].

Ernst *et al.* [8] provides in-depth studies on the scheduling problem. They provide descriptions of scheduling modules as well as references to algorithms used to solve them. They suggest a number of modules associated with the processes of constructing a roster. The development of any particular roster may need only some of these modules. In certain cases, several modules may be combined into one procedure. Ernst *et al.* proposed six general modules: demand modeling, days off scheduling, shift scheduling, line of work construction, task assignment and staff assignment. *Demand modeling* is determining how many workers are needed at different times over some selection period or rostering horizon. *Days off scheduling* is the determination of how off days are to be scheduled between work days. *Shift scheduling* is the problem of selecting what shifts are to be assigned. *Line of work construction* is the creation of work schedule over the time horizon for each worker, depending on the module it can be concerning shifts, duties, or tasks. *Task assignment* is the assignment of one or more jobs to be performed during each shift while considering job requirements such as certain skills or levels of seniority. *Staff assignment* is the assignment of individual staff to the lines of work [8].

The Ernest *et al.* [8] modules provide a general framework within which different rostering models and algorithm may be placed. In their study, they stated three main factors that influence the differences between rostering problems and models:

1. The degree to which days off scheduling, line of work construction and task assignment are integrated.
2. The modules required to construct a roster.
3. The demand and the fundamental unit from which lines of work are constructed.

They provide a brief description of the key problems related to staff scheduling in different application areas. In transportation systems such as airlines, railways, mass transit and buses, the staff scheduling is known as crew scheduling. Each assignment is characterized by its starting time, finishing time, and location. All staff assignments are carried out per a given timetable. There are also several types of transportation operations that require performing tasks with different type of skills at the same location. Examples of this include airport ground operations, cargo terminals and aircraft maintenance personnel. The difficulty in such problems is the allocation of a variety of small tasks to shifts in the presence of skill constraints [8].

Alternatively, scheduling models for call centers, retail and the hospitality industry do not involve a location feature. The nature and the number of tasks that need to be performed is unknown. All that is known is that a workforce requirement pattern exists for the entire chosen time horizon. This feature complicates scheduling and rostering in call centers. In scheduling for health care systems, the major rostering concern is nurse scheduling. The roster must ensure qualified nurses to cover the demand arising from the number of patients in the wards while satisfying work regulations, distinguishing between permanent and temporary staff, ensuring that different shifts are distributed fairly, allowing for leave and days off, and fulfilling employee preferences as much as possible [8].

An important aspect of scheduling protection and emergency services such as police, ambulance, fire and security services is the need to meet predetermined service standards. These standards may be related to response times to attend incidents or the ability to cover different types of incidents with properly trained officers to different types of incidents. Most emergency services have very tightly controlled regulations specifying acceptable patterns of shift work. Similarly, one must meet a minimum standard in manufacturing. In a dynamic production manufacturing environment, it is important to establish production levels for many different items to meet the demand over a given time while keeping the inventories at acceptable levels. The problem arises with the balance between demand and supply is to calculate the manpower requirements for each production period [8].

Governments at all levels operate a large number of labor intensive services in addition to protection and emergency services. Optimized labor scheduling for civic services and utilities intend to improve the services provided by the government and at the same time minimizing the cost. There are added challenges with this as governments tend to have more generous employment conditions. These can severely restrict the flexibility of the rosters [8].

In the financial sector, scheduling is important for banks in the staffing of tellers and back office staff. The main difficulty is the undetermined requirement over each day. It is possible to cope with this variability through the judicious use of part-time staff and overtime. Another quite different problem arises in the scheduling of audit staff. The main complexity arises through the non-homogeneous nature of demand with a variety of audit jobs that have a mix of skill requirements and different locations [8].

In addition Ernst *et al.* review some rostering methods and techniques such as:

1. Demand modeling
2. Artificial intelligence approaches
3. Constraint programming
4. Metaheuristics
5. Mathematical programming approaches

Employee scheduling in general is concerned with the allocation of employees to specific shifts in order to satisfy certain types of work demands based on employees qualifications [2]. Al-Yakoob *et al.* [2] is concerned with the assignment and scheduling of different classes of employees at gas stations owned by the Kuwait National Petroleum Company (KNPC). KNPC owns 86 stations located throughout Kuwait. There are three working shifts with three employee categories: supervisor, cashiers, and service workers. There are two types of stations to be assigned with these employees. The first type is a full service station which requires a supervisor and a number of service workers. The second type is a self service station which requires a supervisor, a cashier, and a number of service workers. In their study, Al-Yakoob *et al.* [2] did not deal with the scheduling of service workers because it is done by another company.

For generality, they assume that a subset of the supervisors can work as cashiers and a subset of the cashiers can work as supervisors. The unavailability of a supervisor in full service station or a cashier in a self service station leads to a temporary closing down of the station and the scheduler is fined by KNPC. During the study the following restrictions were taken into consideration:

1. An employee is not allowed to work during two consecutive shifts.
2. An employee may not work for six consecutive days.
3. A certain number of off-days could be required to coincide with weekends during the time horizon.

They developed a mixed-integer programming model for the problem with the following binary decision variables:

$$w_e = \begin{cases} 1 & \text{if employee } e \text{ is selected in the work schedule during the specific time horizon} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{e,d} = \begin{cases} 1 & \text{if } d \in D_e \text{ where } D_e \text{ is the set of permitting working days for employee } e \\ 0 & \text{otherwise} \end{cases}$$

$$x_{e,g,d,t} = \begin{cases} 1 & \text{if employee } e \text{ is assigned to station } g \text{ on day } d \text{ during shift } t \\ 0 & \text{otherwise} \end{cases}$$

Various constraints were used to formulate the problem:

1. Stations requirements
2. Workloads, and on/off days
3. An employee may work at no more than one station during any given shift

4. Avoiding the assignment of consecutive shifts to employees

5. Comparing relative dissatisfaction levels of employees

The objective function of the proposed employee scheduling model is to minimize:

$$\alpha_1 \sum_{e \in E} w_e + \alpha_2 \sum_{e \in E} \sum_{g \in S_e} \sum_{d \in D_e} \sum_{t \in T} c_{egt}\, x_{egdt} + \alpha_3 \sum_{e \in E} \sum_{d \in D_e} c_{ed}\, (w_e - z_{ed}) + \alpha_4 \sum_{e \in E} v_e$$

It consists of four terms. The first term seeks to use the minimum number of employees. The second and third terms of the objective function aim to minimize the total employees' expressed preferences for specific stations and daily shifts, and for off-days, respectively. The term c_{egt} indicate the desirability of employee e to work at station g during shift t and the term c_{ed} indicate the penalty of assigning employee e on an off day d. The fourth term attempts to achieve fairness in the generated employee schedules by minimizing the total sum of absolute differences between employee preference indices and a central preference value. The weight factors α_1, α_2, α_3, and α_4 reflect the relative significance to the scheduler of achieving the respective pertinent preferences. Because of the size of the model, a two-stage reformulation of the problem was proposed where in the first stage, employees are assigned to stations, and in the second stage, the assigned employees are scheduled to work during shifts on specific days.

Combs [5] discusses the crew scheduling problem and its variants in perspective of airline crew scheduling problem in his dissertation. His objective was to minimize the number of crews required and maximize the efficiency of the crews, subject to the following constraints:

1. Each flight of the aerial refueling problem must be flown uniquely.

2. Crew duty days must not exceed 16 hours.

3. Once a crew duty day is over, a crew must rest for a minimum of 12 hours.
4. Crews can fly no more than 125 hours in 30 days and 330 hours in 90 days.
5. The user-defined minimum waiting time between flights must be met.
6. Bases of arrival and departure must match for each crew and aircraft.

He developed an efficient tabu search [11] approach to Air Mobility Command's Tanker Crew Schedule Problem. In the tabu search algorithm, the first task is creating an initial solution. It then uses a global tabu search approach to find better solutions.

Freling *et al* [10] deals an integrated approach to vehicle and crew scheduling problem (VCSP) for an urban mass transit system with a single depot. VCSP is different from the sequential approach where vehicles are scheduled before scheduling the crews. For evaluation purposes, they also consider the traditional sequential approach and the opposite sequential approach by scheduling crews before vehicles. VCSP is to find feasible and minimum cost schedule for the vehicles and the crews for a set of assignments or trips, while dealing with fixed time horizon, fixed starting and ending times, and the time required between any two locations are known.

For their study, they made three main assumptions. The first was that the vehicle scheduling characteristics correspond to one depot, identical vehicles, and no time constraints. The second was that the cost function for the VCSP is the summation of the vehicle and crew scheduling cost function. Finally, they assumed that the fact whether or not a sequence of tasks on one vehicle block can be performed by a single crew member without interruption only depends on the duration of this sequence.

In their model, they distinguish between two types of tasks. The first is a trip task. The second is the deadhead task, meaning the vehicle is departing the depot or returning to the depot, or the vehicle is outside the depot but it is between two trips.

Although in their model, they define a duty as the tasks that are assigned to the same crew member. The mathematical formulation for the VCSP is a combination of vehicle scheduling and crew scheduling. The vehicle scheduling is consider as quasi-assignment where it is an assignment problem with special condition or an extra constraints, in their case the vehicle will be assigned only if its cover certain jobs after others. The model is governed by the following constraints:

1. Each trip task must cover by one duty
2. Consider the vehicle quasi assignment formulation
3. Guarantee the link between deadhead tasks and deadhead in the solution
4. Guarantee that each deadhead is covered by a duty

The decision variables for the model is as follow:

$$y_{ij} = \begin{cases} 1 & \text{if a vehicle covers trip } j \text{ directly after trip } i \\ 0 & \text{otherwise} \end{cases}$$

$$x_k = \begin{cases} 1 & \text{if duty } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

the objective function of the model seeks to minimize the total cost of vehicle and crew, and it is formulated as follows:

$$\min \sum_i \sum_j c_{ij}\, y_{ij} \;+\; \sum_k d_k\, x_k$$

where the coefficient c_{ij} is corresponding to the vehicle cost between i and j. The coefficient d_k is corresponding to crew cost of duty k. To solve this problem, they used Lagrangian Relaxation with column generation and they were able to obtain a good feasible solution within reasonable computation times.

Bouarab *et al.* [4] studies and analyzes the nurse scheduling process in practice for two different hospitals. They propose models and heuristics to improve both the process and the quality of the resulting schedule. In both designated hospitals, the nurse scheduling is done manually for 28 days by an assigned clerk. For the study, a subset of nurses at Hospital 1 followed a rotation system while others were assigned to one specific shift. For Hospital 2, a float team was selected where all nurses were assigned to a specific shift. In both hospitals, the clerk goes through the following steps to create a schedule:

1. Collect preferences
2. Sketch the schedule
3. Correct the schedule
4. Post the schedule
5. Adjust the schedule

In this case they found that there is no rules to govern the relationship between the clerk and nurses, both will try to reach the win-win relation level, this relation will lead to unfamiliar and personnel rules or constraints which will be difficult to implement and accommodate. The constraints in nurse scheduling are:

1. Demand nurses for each shift, for this constraint there is no estimation of this demand provided, historical data for the last year was traced to evaluate the average demand for each shift and each day as well as the variance.
2. Shifts that can be assigned to each particular nurse

3. Maximum number of consecutive days of work

4. Minimum amount of rest time between two shifts

5. Isolated days of work or days-off

The decision variables for the model

$$x_{ijk} = \begin{cases} 1 & \text{if the nurse } i \text{ is assigned to shift } k \text{ on day } j \\ 0 & \text{otherwise} \end{cases}$$

$$z^+_{jk} = \text{number of nurses over-covering the day } j \text{ in the shift } k$$

$$z^-_{jk} = \text{number of nurses under-covering the day } j \text{ in the shift } k$$

$$M_i = \text{length of the last work sequence in the previous period}$$

$$x_{ijk} \ \in [0, 1]$$

$$z^+_{jk},\ z^-_{jk} \in \mathbb{Z}_{\geq 0} \text{ for } j = 1, \ldots, 28 \text{ and } k = 1, \ldots, \text{number of shifts}$$

The objective function is constituted of three terms. The first one specifies that rotation from one shift to another is minimized. The second one is a quadratic term that ensures that penalty increase rapidly when moving away from the unit requirements and finally the third handle that preferences are maximized. The first term of the objective function of the first hospital model was removed in the objective function of the second hospital because the nurses does not rotate over shifts. The objective function of the model for the first hospital is as follows:

$$\min \sum_{j=1}^{27} \sum_{i \in N} \sum_{k \in K} \gamma_{ik}(x_{ijk} - x_{ij+1,k})^2 + \sum_{j=1}^{28} \sum_{k \in K} \left[c^+ \left[z^+_{jk}\right]^2 + c^- \left[z^-_{jk}\right]^2 - \sum_{i \in N} \beta_i p_{ij} x_{ijk} + r_i f_{ij} x_{ijk} \right]$$

and the objective function of the model for the second hospital is as follows:

$$\min \sum_{j=1}^{28} \sum_{k \in K} \left[c^{+} \left[z_{jk}^{+} \right]^{2} + c^{-} \left[z_{jk}^{-} \right]^{2} - \sum_{i \in N} \beta_i p_{ij} x_{ijk} + r_i f_{ij} x_{ijk} \right]$$

The models were solved by means of optimization software. However, it took a long time to solve the problem to optimality. Solving the problem by heuristic algorithm was faster.

Bard *et al* [3] presents a new methodology for solving the nurse scheduling problem which led to the development of a branch-and-price algorithm. In their study, they start with an initial schedule and attempt to adjust individual work assignments according to the daily fluctuations in the patient population, absenteeism and emergencies. In this case, a scheduler has the option to call nurses in on their day off, use available nurses with overtime pay, use outside resources, pool nurses or to operate with the shortage.

The problem associated with making the daily adjustments is formulated as an Integer Program (IP) and solved within a rolling horizon framework. The idea is to consider 24 hours at a time, but to only implement the results for the first 8 hours. The IP is then resolved for the next 24 hours after several hours have elapsed and new data are available, and so on. Included in the algorithm are a feasibility heuristic to find the upper bounds and a cut generation procedure to improve the lower bound computations. A set-covering-type IP was used to find upper bounds and mixed-integer rounding cuts were used to tighten the relaxed feasible region. Although the effectiveness of all but the set covering heuristic proved to be marginal, most problem instances with up to 200 nurses was solved within 10 minutes on a 1.1 GHz PC.

In formulating the model, they assumed that all cancellation, overtime costs, demand are known for each of the three shifts, and that the status of all nurses is known. This means that the nurse managers, the supervisors, and the nursing resources director all have up-to-date information on call outs, shortages, surpluses, floaters, and pool nurses. They designed their model to reflect the point of view of the hospital and it is intended for use by the nursing services office rather than the unit managers. The decision variables of the model are:

$$x_{ijp} = \begin{cases} 1 & \text{if nurse } i \text{ is assigned to unit } j \text{ in period } p \\ 0 & \text{otherwise} \end{cases}$$

$$z_{jp} = \text{number of agency nurses assigned to unit } j \text{ in period } p$$

$$o_{jp} = \text{number of on-call nurses assigned to unit } j \text{ in period } p$$

$$v_i = \begin{cases} 1 & \text{if nurse } i \text{ (regular or pool)is assigned to a certain shift in the midterm schedule but is not} \\ & \text{needed (shift is cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ip} = \begin{cases} 1 & \text{if nurse } i \text{ is given a split assignment (regular or overtime) for a shift starting in period } p \\ 0 & \text{otherwise (a 12-hour shift to be split between two or three units)} \end{cases}$$

$$g_{jp} = \text{number of gaps (shortages) in unit } j \text{ during period } p$$

The model is governed by constraints such as:

1. Demand must be met
2. Restriction on nurses floating between unit due to qualification and preferences
3. Prohibit the assignment of overtime without a regular assignment for regular nurses who either float or work overtime
4. Track the occurrence of split shift, where a nurse will be assigned in a shift in

two different unit at a same time.

5. Ensure a nonconsecutive assignment

6. Ensures that the overtime assignments are non-increasing

7. Account for preference violations in the adjusted schedule

The model objective function is as follows:

$$\min \sum_{i\in R\cup P}\sum_{j\in j(i)}\sum_{p\in T(i)\cup\overline{\mathrm{T(i)}}} c^{1}_{ijp}x_{ijp}+\sum_{i\in R\cup P} c^{2}_{i}v_{i}+\sum_{j\in J}\sum_{p\in T} c^{3}_{jp}o_{jp}+\sum_{j\in J}\sum_{p\in T} c^{4}_{jp}z_{jp}+m\sum_{j\in J}\sum_{p\in T} g_{jp}$$

The objective function sums the costs of each alternative available for handling shortages. The first term represents the cost of assigning nurse i to unit j in period p, the second term represents the cost of assigning an unneeded nurse, the third term represents the cost of assigning on-call nurses to unit j in period p, the fourth term represents the cost of assigning agency nurses to unit j in period p and the fifth term penalizes under coverage when insufficient internal or external resources are available. The overall goal was to satisfy coverage requirements at a minimum cost while taking into account nurse preferences, morale, the need for the perception of fairness, and the expected response of staff members whose work patterns are affected. The computational result showed that by using branch and price algorithm the daily adjustment can be solved efficiently for up to 200 nurses.

Pinedo [13] outlines some of the most common problems encountered in scheduling. Empirically, the problems that are relevant to resource scheduling environments are summarized by Pinedo as theoretical models that deal with the problem of n jobs required to be scheduled. The problem is solved after scheduling these n jobs. In reality, regularly there will be a new jobs to be added or an existing jobs to be rescheduled. The dynamic features of personnel scheduling in services may require considering and implementing emergences or unexpected events in the schedule.

Theoretical models usually do not assure the resequencing problem. In practice, the following problem often occurs: There exists a schedule, which was developed based on certain restrictions, and an unexpected event occurs which leads to either major or minor changes in the existing schedule. The rescheduling process, which is also referred to as reactive scheduling, may have to satisfy certain constraints. For example, one may wish to keep the changes in the existing schedule at a minimum even if an optimal schedule cannot be obtained. This implies that it is advantageous to construct schedules that are robust and resilient to change.

Real world scheduling environments are often more complicated than the ones considered in general scheduling theory. In the mathematical models, the priorities of the tasks are typically assumed to be fixed (*i.e.* they do not change over time). In practice, the priorities of a task often change over time due to changing priorities in the organization or a number of other factors. Some mathematical models do not take preferences into consideration. Some assignment may be favored over others for reasons that cannot be integrated into the model. Most theoretical research is concerned with models featuring a single objective. Most real world problems have multi-criteria and multi-objective features, which sometimes are in conflict with each other [13].

Pinedo states that “scheduling, as a decision-making process, plays an important role in most manufacturing and production systems as well as in most information-processing environments” [13, P.1]. A general personnel scheduling system should be driven by three key factors: Effectiveness, so that customer expectations are met; Efficiency, so that employee needs are met; and Efficacy, so organizational objectives are met and because of the variety of requirements, work rules, and objectives that occur in a specific personnel scheduling situation each problem may appear unique.

However, there are often common elements and problems and each of these problem areas must be dealt with in any personnel schedule [12]. the common element are as follow:

1. Days on, days off, and work rosters
2. Shift assignments and work rosters
3. Tours of longer than a standard working day
4. Integrated work week assignment of tours and days off
5. Meal breaks and rest periods
6. Part-time employees
7. Paid non-working days (vacation, sick leave, training, meetings, off sites)
8. Overtime and absences
9. Remuneration (wages, premiums, and so forth)
10. Alternative work patterns

Causmaecker *et al.* [6] provides a classification of real world personnel scheduling problems. The authors visited eleven different companies and interviewed them about their experience with personnel scheduling. Causmaecker *et al.* identified the following important parameters for the personnel planning problem:

1. The number of available employees,
2. The amount of overtime,
3. Fixed or fluctuating demands,

4. The work contracts,
5. The robustness of the planning,
6. The possibility of employees to mutually swap tasks,
7. The flexibility of the schedule.

They also mention several parameters to determine the criteria and quality measures of a good schedule: the quality of the schedule for the individuals, the quality of the schedule for the organization, the difference between schedule and execution, the incorporation of personnel preferences, the impact of a change, and the cost of executing a schedule.

2.3 Summary

This chapter reviewed previous relevant research about personnel scheduling procedures to see how similar problems were modeled and solved. In the following chapter, a solution technique, scheduling algorithm, and developed scheduling model will be discuss.

III. Methodology

3.1 Chapter Overview

The chapter present the research methodology for the 28th Operational Weather Squadron scheduling problem. This chapter begins with a description of current 28th OWS scheduling process to provide better insight about the study. After that, objectives and restrictions of the 28th OWS schedule model are explained. Following that, elements and notations which compose the schedule are discussed along with their definitions and an integer formulation representing the problem ar introduce. Finally, a MATLAB® and IBM ILOG CPLEX® flow chart illustration is introduced to show how the proposed system is designed.

3.2 28th OWS Scheduling Process

One or more SNCOs spend hours every month manually generating the duty schedule. The large number of requests and requirements make it difficult to generate a satisfactory and error-free schedule using this method. Before generating the schedule, all required information is gathered from each flight. Personnel on leave, TDY, training, and any similar data is collected to generate the schedule. After gathering all of the required information, the scheduler then attempts to match each person with the suitable job taking into consideration all job restrictions and unit regulations. While schedulers assign a person to a job, there are an array of details to be consider, such as availability, qualifications, flight shift cycle, days on and days off, number of consecutive working and off days, trying to schedule each flight in same shift as much as possible, and trying to maintain the minimum number of working or off days for each persons. Since this process is performed manually, it requires an inordinate effort and can take a long time to produce a feasible schedule.

3.3 Programming Language

The selection of the correct programming language must be based on their availability as well as ease to learn and use. The required software has to be capable to solve large problems in effective time. The 28th OWS scheduling problem has more than 62,000 variables and more than (300) constraints. The model was implemented and solved on MATLAB® software. MATLAB® is intended primarily for numerical computing. It is a multi-computing environment. MATLAB® allows matrix functions, plotting of functions and information, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages. MATLAB® can call functions and subroutines written in other languages for other software such as IBM ILOG CPLEX®. CPLEX® connector for MATLAB® gives a high performance interactive environment for solving mathematical optimization problems. CPLEX® Optimizer solves integer programming problems, very large linear programming problems and different types of quadratic programming problems [15].

3.4 Objectives and Restriction

The aim of this study was to develop a robust but effective approach to generate the required schedule. The approach will consider assigning each person to a certain job during each operating shift while complying with the job qualification restriction and requirements. There are eight different jobs that need to be fill by qualified personnel: Site Support, Flight Team Lead (FTL), Flight Weather (WX) Briefer, Joint Operating Area Forecast (JOAF), Hazards, Local Area Weather Chart (LAWC) Synoptician, Regional Weather Supervisor Graphics (RWSG) and Senior Duty Officer (SDO). Each job has a preferred rank of the person to be assigned to it. Each job also has a minimum required number of personnel to perform the job during each shift.

The approach in this thesis attempts to minimize the number of personnel assigned to each job.

There were some restrictions taken into consideration. These restrictions are related to unit regulation and job requirements. The restrictions are as follows:

1. Personnel must be qualified to perform the jobs to which they are assigned.
2. No one can be assigned to work more than one shift in any 24 hour period.
3. No one can work more than a predetermined number of days in a row.
4. Each person must have a predetermined number of days off per week.
5. No one works more than one job per shift.
6. No one gets more than a predetermined number of sequential days off at a time.
7. Personnel must have more than one day off in a row.
8. Personnel cannot be scheduled for a single work day in between days off.
9. Ensure there is enough people assigned to each job for each shift.
10. Keep personnel from the same flight on the same shift as much as possible.

All restrictions should be considered while building the model. The unit may add other restrictions or ignore an existing restriction.

3.5 Scheduling Algorithm

28th OWS schedule production follows a sequence of seven steps. They are:

1. Preferences Gathering.
2. Flight information update.

3. Selecting time horizon and working pattern.
4. Produce the schedule.
5. Reviewing the produced scheduled.
6. Fixing the schedule using the software or manually if necessary.
7. Finalize and confirm the schedule.

The first two steps occur before starting the scheduling process. Before beginning, it is critical to update the availability of each flight member, personnel rank changes due to any promotions, and qualification changes due to training or decertification. The third step is to choose the schedule time period. This will give one the ability to divide the normal time horizon of 28 days to a shorter or longer interval. The length of schedule chosen also determines the working pattern, meaning how many working days and off days each person has in each week. Step 4 is the purpose of the design model.

Steps 5 and 6 are only necessary in cases where a large number of personnel are unavailable for whatever reason. It also provide an opportunity to utilize non-quantifiable command knowledge. In the designed model, there is a step to check errors in the produced schedule. In the event that it is not possible to generate a feasible schedule, the scheduler will have to alter the condition by either changing the working pattern, time horizon, or by including a missing person in the model to yield a feasible schedule. The scheduler can then try to replace the added person with best choice of replacement manually. Step 7 is to confirm the output schedule and distribute it to the flights.

3.6 Preferences

There are two sets of distinct preferences established beforehand by policy that must be accounted for in the final 28th OWS schedule. The first involves assigning personnel to the position(s) most appropriate for their rank and experience whenever possible. The second set of preferences involves keeping personnel in the same flight together on the same shift as much as possible.

Job Preference

Each job has a preferred rank for the person assigned to perform the job. The designed model will try first to schedule jobs to the preferred rank, but in cases where personnel in the preferred rank are not available model will assign someone who is qualified with the next preferred rank.

Flight Preference

Squadron leadership prefers to schedule members of each flight to work together on the same assigned shift whenever possible. This makes it easier for the flights to manage TDYs, training, and leave schedules. It also facilitates unit cohesion within the flight, thereby increasing unit capability. This preference is only disregarded in cases where there an insufficient number of qualified personnel available to meet all requirements for a particular shift.

3.7 Model

A BILP formulation was develop to solve the 28th OWS personnel scheduling problem. The notation and decision variables in the model are discussed in the following sections.

Sets and Matrices

A list of sets and matrices used in the model formulation is provided below:

ω : number of working days in a week, define as *numon* in the MATLAB® code.

$\nu = 7 - \omega$: number of days off in a week.

S : Set of squadron personnel. $|S|$ is *numppl* in the MATLAB® code.

M : Set of military rank values for each member of the squadron. Model values are assigned from actual pay grade as described in Table 1.

W : Set of weeks in the schedule. $|W|$ is *numweek* in the MATLAB® code.

P : Set of jobs to fill on each shift. $|P|$ is *numjob* in the MATLAB® code.

R : Set of preferred ranks for each job position. Model values are assigned from actual pay grade as described in Table 1.

F : Set of flights. Same as shifts per day. $|F|$ is *numflt* in the MATLAB® code.

Q : $|F| \times |W|$ matrix such that for all $q_{fw} \in Q$, q_{fw} = the preferred shift for flight f on week w.

Table 1. Model Military Ranks

Pay Grade	**Military Rank in Model**
E-1 to E-4	1
E-5	2
E-6	3
E-7	4
E-8	5
E-9	6
O-1 to O-2	7
O-3	8
O-4	9
O-5	10

D: The time horizon. $|D| = 7 * |W|$ is *numday* in the MATLAB® code.

N : $|P| \times |F|$ matrix such that for all $n_{pf} \in N$, n_{pf} = number of personnel required to fill job p on shift f.

G : $|S| \times |P|$ matrix such that for all $g_{sp} \in G$, $g_{sp} = \begin{cases} 1 & \text{if person } s \text{ is qualified to perform job } p \\ 0 & \text{otherwise} \end{cases}$

Decision Variables

Definition of the binary decision variables is as follow :

$$x_{sdfp} = \begin{cases} 1 & \text{if person } s \text{ working on day } d \text{ and shift } f \text{ is assigned job } p \\ 0 & \text{otherwise} \end{cases}$$

These decision variables were chosen in this manner for two reasons. The first reason is that this construction gives more flexibility to the scheduler by permitting job changes during a period of work days. The second reason is that this construction provides convenient formulations for the model constraints. Furthermore, while there may appear to be an unnecessarily large number of variables, in practice, the G matrix is sparse, meaning most of these decision variables are deleted in pre-processing.

Constraints

The various constraints enforced by the squadron employee scheduling problem are presented below:

Shift restrictions

$$\sum_{p \in P} \sum_{i=f}^{|F|} x_{sdip} + \sum_{i=1}^{f-1} x_{s(d+1)ip} \leq 1 \qquad \forall\ f \in F - \{1\},\ s \in S \text{ and } d \in D - \{|D|\} \tag{1}$$

$$\sum_{f \in F} \sum_{p \in P} x_{sdfp} \leq 1 \qquad \forall\ s \in S \text{ and } d \in D \tag{2}$$

Constraints (1) and (2) ensure that each person will be assigned only work one shift for any 24-hour period.

On / Off days restrictions

$$\sum_{i=d}^{d+\nu}\sum_{f\in F}\sum_{p\in P} x_{sifp} \geq 1 \qquad \forall\, s\in S \text{ and } d\in\{1,2,\ldots,|D|-\nu\} \tag{3}$$

$$\sum_{i=d}^{d+\omega}\sum_{f\in F}\sum_{p\in P} x_{sifp} \leq \omega \qquad \forall\, s\in S \text{ and } d\in\{1,2,\ldots,|D|-\omega\} \tag{4}$$

$$\sum_{f\in F}\sum_{p\in P} x_{s(d-1)fp} + x_{s(d+1)fp} - x_{sdfp} \leq 1 \qquad \forall\, s\in S \text{ and } d\in D-\{1,|D|\} \tag{5}$$

$$\sum_{f\in F}\sum_{p\in P} x_{sdfp} - x_{s(d-1)fp} - x_{s(d+1)fp} \leq 0 \qquad \forall\, s\in S \text{ and } d\in D-\{1,|D|\} \tag{6}$$

$$\sum_{i=d}^{d+6}\sum_{f\in F}\sum_{p\in P} x_{sifp} \leq \omega \qquad \forall\, s\in S \text{ and } d\in\{1,8,\ldots,|D|-6\} \tag{7}$$

Constraint (3) ensures that no one is assigned more than a certain number of days off. This is decided by the scheduler according to the working pattern. Constraint (4) ensures that no one works more than ω days in a row. Constraint (5) ensures that no one is scheduled for a single, non-consecutive day off. Constraint (6) ensures that no one is assigned a single, non-consecutive working day. Constraint (7) ensures every person will work no more than ω days per week.

The next set of constraints ensure that the squadron preferred on/off work day patterns are selected as much as possible. Table 2 lists the preferred work patterns for a single week with a four days on/three days off schedule. Table 3 lists the preferred work patterns for a single week with a five days on/two days off schedule. Let $I=\{1,2,\ldots,7\}$ and $\{H_i : i\in I\}$ be a family of sets such that $j\in H_i\subset I$ if an individual must be scheduled for either day i or day j in all weekly schedule choices listed in either Table 2 or Table 3, depending on the selected value of ω. Constraints (8) then enforce the on/off days patterns shown in Tables 2 or 3.

$$\sum_{f\in F}\sum_{p\in P} x_{s,7(w-1)+i,f,p} + \sum_{j\in H_i} x_{s,7(w-1)+j,f,p} \geq 1 \qquad \forall\, s\in S,\ w\in W \text{ and } i\in I \tag{8}$$

Table 2. Single Week Working Pattern of $\omega = 4$ Days Working and 3 Days Off

day pattern	1	2	3	4	5	6	7
1	on	on	on	on	off	off	off
2	off	on	on	on	on	off	off
3	off	off	on	on	on	on	off
4	off	off	off	on	on	on	on
5	on	off	off	off	on	on	on
6	on	on	off	off	off	on	on
7	on	on	on	off	off	off	on

Table 3. Single Week Working Pattern of $\omega = 5$ Days Working and Days Off

day pattern	1	2	3	4	5	6	7
1	on	on	on	on	on	off	off
2	off	on	on	on	on	on	off
3	off	off	on	on	on	on	on
4	on	off	off	on	on	on	on
5	on	on	off	off	on	on	on
6	on	on	on	off	off	on	on
7	on	on	on	on	off	off	on

Job restrictions

Constraints (9) ensure that at least the minimum number of qualified personnel required is assigned to each job for each shift in the schedule.

$$\sum_{s \in S} g_{sp}\, x_{sdfp} \geq n_{pf} \qquad \forall\, d \in D, f \in F \text{ and } p \in P \tag{9}$$

Objective Function

The 28th OWS personnel scheduling model seeks to ensure a sufficient number of qualified personnel are assigned to each duty position for each shift on the schedule in a manner that places personnel in their primary jobs as often as possible and in secondary positions only when necessary. The model also seeks to keep members of the same flight together on the same shift as often as possible. Thus, the objective function values are:

$$c_{sdfp} = \begin{cases} g_{sp} \cdot \min\{M_s, R_p\} & \begin{array}{l}\text{if person } s \text{ is a not a member of the flight} \\ \quad \text{preferred for shift } f \text{ on day } d \text{ for } p \in P\end{array} \\ \\ g_{sp} \cdot \left(\min\{M_s, R_p\} + \max\limits_{s \in S}\{M_s\} \right) & \begin{array}{l}\text{if person } s \text{ is a member of the flight} \\ \quad \text{preferred for shift } f \text{ on day } d \text{ for } p \in P\end{array} \end{cases}$$

Which yields an objective function for the model:

$$\max \sum_{s \in S} \sum_{d \in D} \sum_{f \in F} \sum_{p \in P} c_{sdfp} \ x_{sdfp} \tag{10}$$

Since the objective function is maximizing, the model will tend to choose the max weight c_{sdfp} as much as possible. It will schedule flight members to the preferred shift first on each day. The personnel qualification factor g_{sp} will ensure only qualified personnel are assigned to each position. Finally, choosing the minimum between the military rank and the job preferred rank ensures scheduling personnel to the job corresponding to their preferred rank as much as possible.

3.8 Model Flow Chart

Figure 1 illustrates a flow chart of the 28th OWS scheduling process. The process starts with data collection. Then, the scheduler decides time horizon and the working pattern. Later, the model is run and output is produced. If a feasible schedule is found, the output schedule will be confirmed. If the output is infeasible, the schedule model output will need manual adjustment until a feasible schedule is produced.

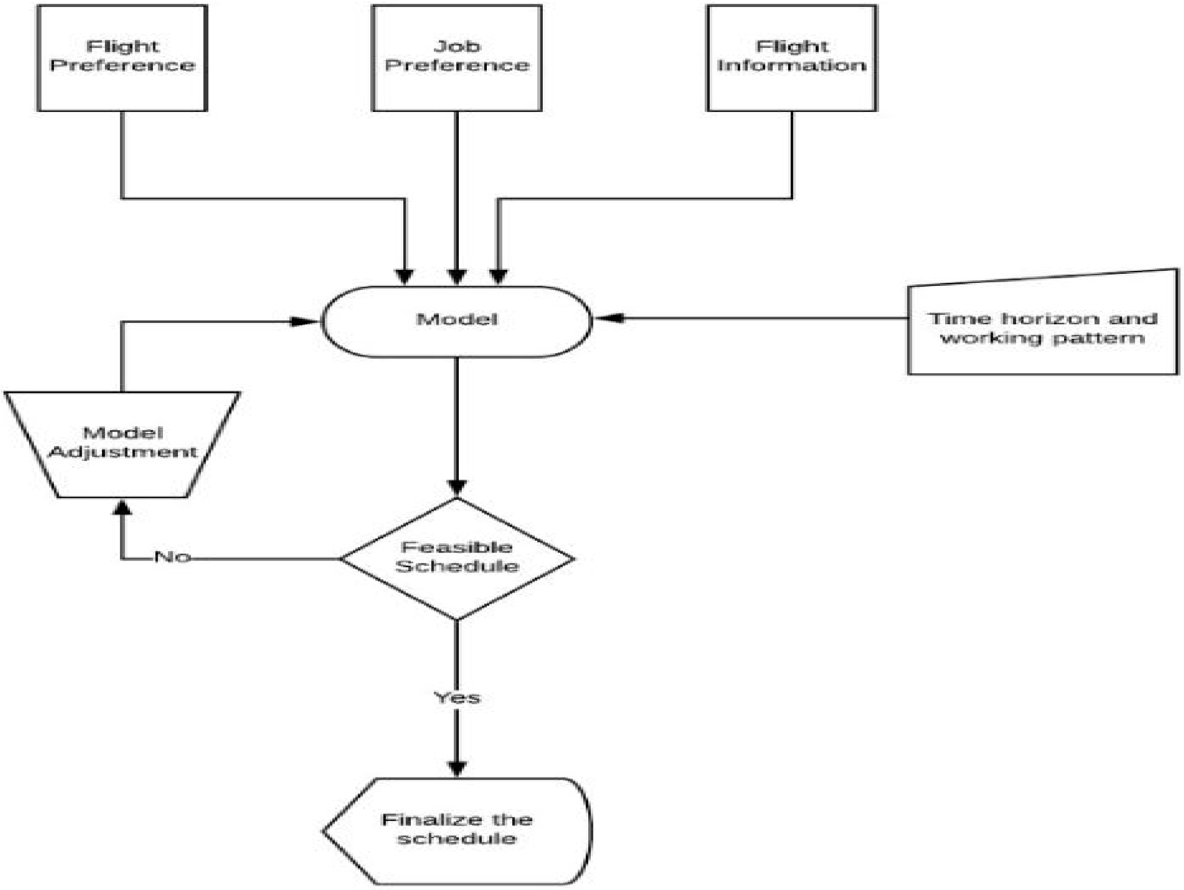

Figure 1. Model Flow Chart

3.9 Summary

In this chapter, a description of 28th OWS personnel scheduling process was given. The 28th OWS scheduling model objective and restrictions were discussed. Following that, the scheduling algorithm was introduced in detail. Finally, a BILP formulation was depicted along with notation, decision variables, constraints, and the objective function was explained. In the next chapter, an analysis is conducted with the model is reviewed. Model performance is also reviewed.

IV. Analysis

4.1 Chapter Overview

In this chapter, an analysis of the applied solution technique is examined. First, the performance of the scheduling models response time is evaluated in different scenarios. Then, different required input adjustments to overcome the infeasibility will be explained. The scheduler preferences satisfaction will then be checked. Finally, the contribution of the developed scheduling model will be discussed.

4.2 Performance of Scheduling Model

Different likely scenarios must be examined to measure the performance of the model under realistic conditions. One of these measures is the time required for the scheduling model to run and produce an output. It is important to develop a feasible schedule in reasonable time.

The MATLAB® 28th OWS personnel scheduling model was solved using only the aforementioned constraints on an Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz processor and 64.0 GB RAM. No additional restrictions for particular days off, vacations, outside training, and so forth. were present in the model. The MATLAB® internal integer linear program solver produced a feasible schedule after approximately seven hours. This time is considerably better than the manual approach. However, it is a very long time for the intent of this study. For this reason, the IBM ILOG CPLEX® mixed integer linear program solver function was called through MATLAB® to solve the scheduling problem. This produced a feasible schedule in less than two minutes, which is a much more suitable time for day-to-day use.

The scheduling model was next ran 16 times using the CPLEX® add-in with different scenarios to evaluate model performance. The number of available personnel, time horizon, working pattern, and number of flights were changed to examine solution time of the model in each scenario. The notional scenarios are:

1. All personnel are available to work any job they are qualified for during any shift across the time horizon.
2. Some personnel are completely unavailable and the remaining personnel are available for the entire time horizon to perform the assigned job for the selecting time horizon and working pattern.
3. One entire flight is missing and personnel from the other two flights members are available and able to perform the assigned job for the selected time horizon and working pattern.
4. One entire flight is missing and some of the personnel from the other two flights are unavailable. The remaining personnel are available and able to perform the assigned job for the selecting time horizon and working pattern.

Personnel missing for all scenarios are fixed and arbitrary chosen. The missing flight was chosen as the biggest flight with the largest number of personnel qualified over all jobs to highlights the scenario's impact.

The results are listed in Table 4. Some scenarios resulted in an optimal schedule where all the 28th OWS scheduling model restrictions were met. In other scenarios, the the 28th OWS scheduling model was unable to produce an operational feasible schedule. These scenarios required manual intervention to develop an operational schedule. However, generating a feasible schedule manually by a human scheduler takes more than one working day. Thus, even under these conditions, the 28th OWS personnel scheduling model is still faster than pure manual approach.

Table 4. Model Performance Table

scenario	28 days/ 4 working days	28 days/ 5 working days	14 days/ 4 working days	14 days/ 5 working days
1	feasible schedule 97.69 seconds	feasible schedule 34 seconds	feasible schedule 6.89 seconds	feasible schedule 6.55 seconds
2	infeasible schedule 7 seconds	feasible schedule 96.69 seconds	infeasible schedule 5 seconds	feasible schedule 7.59 seconds
3	infeasible schedule 7 seconds	infeasible schedule 6 seconds	infeasible schedule 5 seconds	infeasible schedule 5 seconds
4	infeasible schedule 7 seconds	infeasible schedule 7 seconds	infeasible schedule 5 seconds	infeasible schedule 6 seconds

4.3 Adjustments

The model was able to efficiently solve the first scenario with different time horizons and working patterns. In the second scenario, a feasible schedule was found by changing the working pattern to $\omega = 5$ and then deleting surplus scheduled days to get the minimal required number of personnel on each shift. In the third and fourth scenarios, the model was unable to produce feasible schedules in any situation with different time horizons and working patters. To overcome this issue, the scheduler should introduce missing personnel to the schedule and then try to replace them with the most suitable available replacement. It was found that the Regional Weather Supervisor Graphics position, which it is represented as job 7, had the minimum number of qualified personnel to perform this job. Table 5 shows the total qualified personnels for each flight for each job.

Table 5. Qualified Personnel in Each Flight

Flight	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8
flight A	27	12	11	7	7	8	3	6
flight B	23	8	7	7	6	7	1	3
flight C	24	10	7	8	7	7	2	5
Total	74	30	25	22	20	22	6	14

By adding the qualified personnel for job 7 from the missing flight to the schedule, the model was able to overcome the infeasibility in scenarios 3 and 4. A feasible schedule was produced only for 14 day time horizon and working pattern of $\omega = 5$ days working with two days off. Table 6 shows the result and response time for both scenarios.

Table 6. Adjusting Model Performance Table

scenario	28 days/ 4 working days	28 days/ 5 working days	14 days/ 4 working days	14 days/ 5 working days
3	infeasible schedule 7 seconds	-	infeasible schedule 7 seconds	feasible schedule 13.41 seconds
4	infeasible schedule 8 seconds	-	infeasible schedule 9 seconds	feasible schedule 16.1 seconds

Scenarios two and four with different time horizons and working patterns represent more realistic world problems. The model was able to solve these two problems in effective time.

In general, to overcome infeasibility, the scheduler should follow the following steps until a feasible schedule is produced:

1. Change the time horizon.
2. Change the working pattern.
3. Add missing personnel qualified for job 7.
4. Add missing personnel qualified for job 8.
5. Add arbitrary personnel until feasibility is achieved.

4.4 Preferences

Job and shift preferences are essential information that need to be appropriately considered in the scheduling process. The four scenarios with different time horizons and working patterns were examined to check the quality of the produced schedule. The stability of the assigned jobs during time horizons were also checked. Table 7 shows the model's ability to produce a schedule with stable job assignment for each person in different scenarios all along the decided time horizon.

Table 7. Model Ability Table

scenario	28 days/ 4 working days	28 days/ 5 working days	14 days/ 4 working days	14 days/ 5 working days
1	64.5%	64.5%	72.04%	66.67%
2	-	55.95%	-	60.71%
3	-	-	-	46.3%
4	-	-	-	50.82%

The model was able to produce an output schedule in different scenarios achieving the flight preferences for all available feasible output schedules.

4.5 Contribution of Scheduling Model to Air Force

There are many weather squadrons or units with similar working environments and regulations. It is beneficial to estimate the total labor hours spent on current manual scheduling processes to determine the contribution of the 28th OWS scheduling model in this thesis. From this point of view, the total labor hours spent to produce a feasible schedule will be calculated and compared with the total time required by the model to produce a feasible schedule[7].

Before starting the scheduling time horizon, two to three expert personnel work at least one day on creating the current schedules. This takes three shifts for a total of at least 24 man-hours to produce a schedule which is most often feasible, but not necessarily optimal. For each squadron, the labor hours required to produce a feasible schedule at a time is between 48 and 72 man-hours. In general, squadrons will fix the time horizon to four weeks, which is approximately one calendar month. When we consider yearly totals, the required man-hours is estimated to be between 576 and 864 man-hours, assuming that there are 30 weather squadron or units with similar working environments and regulations in the Air Force. In the case of a four week time horizon, the yearly total man-hours is between 17,280 and 25,920 man-hours. The developed model requires at most two minutes to produce an optimal feasible schedule where possible on a commonly available computer with the required software installed. This means that for a four week time horizon, the yearly total required time for the developed model to produce an optimal feasible schedule is 24 man-minutes. If implemented across 30 units, the yearly total required scheduling time is at most 12 man-hours. The contribution of the developed scheduling model is obvious in this aspect.

As shown in the previous analysis, the developed scheduling model achieves the objective in a very short time relative to the manual methods currently in use. Furthermore, the 28th OWS personnel scheduling model's output will be an optimal schedule satisfying all preferences when solved to optimality while the manual approach will only be guaranteed to be a feasible schedule. Finally, with the 28th OWS scheduling model's assistance, scheduling can be perform by anyone in the unit with minimal training on how to operate the model, as it does not require any understanding of each job requirement or knowledge of personnel qualifications.

4.6 Summary

In this chapter, performance of the developed model was examined by several measures, response times of the model with different scenarios were evaluated. The ability to overcome the infeasibility by manual intervention was discussed. Model preferences satisfaction . Finally, contribution of the developed model is explained. In next chapter, conclusion of research and future recommendation will be discussed.

V. Conclusions and Recommendations

5.1 Chapter Overview

This chapter presents a summary of the research and the conclusions resulting from it. In addition, recommendations for future studies on operational weather squadron scheduling is introduced.

5.2 Summary of Research

In this thesis, the general mission of the 28th OWS was explained followed by the importance of scheduling as well as the challenges associated with creating an optimal schedule efficiently. Next, the research question was determined as: "How can an optimum scheduling for the 28th OWS be created using readily available software and equipment?"

Previous research on crew scheduling was reviewed to learn solution techniques that could be exploited in this research. The research proceeds with illustrations of methodologies applicable to the 28th OWS scheduling problem. The current 28th OWS scheduling process was explained to give better insight into the need for this research. Following that, objectives and restrictions of the 28th OWS schedule model were discussed. Moreover, model elements, notations, and decision variables with their definitions were explained. A BILP formulation representing the problem was then introduced. The applied solution technique was evaluated and demonstrated to be superior to current procedures to the current scheduling methodology across several measures. Finally, the contribution of the 28th OWS scheduling model to Air Force was investigated.

5.3 Conclusion of Research

This study has shown several results about 28th OWS scheduling problem and its applied solution technique. One of these result that the solution technique used in this research and the implemented model can be used in practice for the 28th OWS scheduling and for any weather squadron or any unit with similar working environment and regulations.

The second result is that optimal scheduling can be generated in very short time by the developed model for different scenarios, even in case of infeasibility the model will provide a base start for feasible schedule.

Another result is that choosing MATLAB® as programming language with CPLEX® function to compute the squadron schedule is a proper decision, the performance tests of the developed model confirm that.

Last but not least, with the aid of the developed program a manual intervention occurred. Due to squadron mission and tasks the squadron will miss many peoples which will required changing the main input or manually fixing the schedule.

5.4 Recommendations for Future Research

Further studies in 28th OWS scheduling might investigate heuristics approach to reduce number of variables and constraints instead of direct method. In addition, after applying heuristics another programming languages might be applied such as MATLAB® without CPLEX® function, Java, and Visual Basic for Applications (VBA). As next step of th 28th OWS scheduling, generating of training scheduling along with the crew scheduling, where the developed program will decide who is going to training and when he is going to avoid any conflict with the crew scheduling.

Appendix A. MATLAB Schedule Model

owsmodel.m

```
clear;
clc;

% Add cplex to the MATLAB path
addpath( 'I:\cplex\matlab\x64_win64');

tic;

% Number of weeks in the schedule
numweek = 4;

% Number of days on in a week
numon = 4;

% Positions to be filled on each shift
P = {'Site Support', ...
      'Flight Team Lead (FTL)', ...
      'Flight WX Briefer', ...
      'JOAF', ...
      'Hazards', ...
      'LAWC/5-lvl Synoptician', ...
      'Regional Weather Supervisor Graphics (7 Level or higher only) (RWSG)', ...
      'Senior Duty Officer (SDO)'
};

```

```
% Preferred ranks for each job position
R = [1,3,1,1,3,2,4,8];

% Number of personnel required per shift
N = [4,2,2,ones(1,5)];

% Squadron personnel
S = {'Capt','personnel1','A'; ...
     'MSgt','personnel2','A'; ...
     '2d Lt','personnel3','A'; ...
     '2d Lt','personnel4','A'; ...
     '2d Lt','personnel5','A'; ...
     'MSgt','personnel6','A'; ...
     'TSgt','personnel7','A'; ...
     'TSgt','personnel8','A'; ...
     'SSgt','personnel9','A'; ...
     'SSgt','personnel10','A'; ...
     'SSgt(s)','personnel11','A'; ...
     'SrA','personnel12','A'; ...
     'SrA','personnel13','A'; ...
     'SrA','personnel14','A'; ...
     'SrA','personnel15','A'; ...
     'A1C','personnel16','A'; ...
     'A1C','personnel17','A'; ...
     'A1C','personnel18','A'; ...
     'A1C','personnel19','A'; ...
     'A1C','personnel20','A'; ...
     'A1C','personnel21','A'; ...
```

```
    'A1C','personnel22','A'; ...
    'A1C','personnel23','A'; ...
    'A1C','personnel24','A'; ...
    'A1C','personnel25','A'; ...
    'A1C','personnel26','A'; ...
    'SSgt','personnel27','A'; ...
    'SSgt','personnel28','A'; ...
    'SSgt','personnel29','A'; ...
    'Capt','personnel30','B'; ...
    'MSgt','personnel31','B'; ...
    '2d Lt','personnel32','B'; ...
    '2d Lt','personnel33','B'; ...
    '2d Lt','personnel34','B'; ...
    'TSgt','personnel35','B'; ...
    'TSgt','personnel36','B'; ...
    'TSgt','personnel37','B'; ...
    'SSgt','personnel38','B'; ...
    'SSgt(s)','personnel39','B'; ...
    'SrA','personnel40','B'; ...
    'SrA','personnel41','B'; ...
    'SrA','personnel42','B'; ...
    'SrA','personnel43','B'; ...
    'SrA','personnel44','B'; ...
    'A1C','personnel45','B'; ...
    'A1C','personnel46','B'; ...
    'A1C','personnel47','B'; ...
    'A1C','personnel48','B'; ...
    'A1C','personnel49','B'; ...
```

```
    'A1C','personnel50','B'; ...
    'A1C','personnel51','B'; ...
    'A1C','personnel52','B'; ...
    'A1C','personnel53','B'; ...
    'Amn','personnel54','B'; ...
    'TSgt','personnel55','B'; ...
    'SSgt','personnel56','B'; ...
    'Capt','personnel57','C'; ...
    'MSgt','personnel58','C'; ...
    '2d Lt','personnel59','C'; ...
    '2d Lt','personnel60','C'; ...
    '2d Lt','personnel61','C'; ...
    'MSgt(s)','personnel62','C'; ...
    'TSgt','personnel63','C'; ...
    'TSgt','personnel64','C'; ...
    'SSgt','personnel65','C'; ...
    'SrA','personnel66','C'; ...
    'SrA','personnel67','C'; ...
    'SrA','personnel68','C'; ...
    'SrA','personnel69','C'; ...
    'SrA','personnel70','C'; ...
    'SrA','personnel71','C'; ...
    'A1C','personnel72','C'; ...
    'A1C','personnel73','C'; ...
    'A1C','personnel74','C'; ...
    'A1C','personnel75','C'; ...
    'A1C','personnel76','C'; ...
    'A1C','personnel77','C'; ...
```

```
    'A1C','personnel78','C'; ...
    'A1C','personnel79','C'; ...
    'Amn','personnel80','C'; ...
    'SSgt','personnel81','C'; ...
    'A1C','personnel82','C'; ...
    'TSgt','personnel83',''; ...
    'TSgt','personnel84',''; ...
    'TSgt','personnel85',''; ...
    'SSgt','personnel86',''; ...
    'SSgt','personnel87',''; ...
    'Mr.','personnel88',''; ...
    'Mr.','personnel89',''; ...
    'Mr.','personnel90',''; ...
    'Mr.','personnel91',''; ...
    'Mr.','personnel92',''; ...
    'Mrs.','personnel93','' ...
    };

% Build list of flights in the squadron
F = unique(S(:,3));
F = F(~strcmp(F,''));

% Number of flights
numflt = max(size(F));

% Schedule of which flights are supposed to work which shifts
% Rows of matrix are flight, columns are week and entries are shift
```

```
Q = [1,2,3,1;
     2,3,1,2;
     3,1,2,3];

% Personnel rank
C =
    [8;4;7;7;7;4;3;3;2;2;2;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;2;2;2;8;4;7;7;7;...
     3;3;3;2;2;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;3;2;8;4;7;7;7;4;3;3;2;1;1;
        ...
     1;1;1;1;1;1;1;1;1;1;1;1;1;2;1;3;3;3;2;2;7;7;7;7;7;7];

% Personnel in row i can fill position in column j if entry is
    1, zero otherwise
G= [1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1
    1 1 1 1 1 1 1 1
    1 0 0 0 0 0 0 0
    1 0 0 0 0 0 0 0
    1 1 0 0 0 0 0 1
    1 1 0 0 0 0 0 1
    1 1 0 0 0 0 0 1
    1 1 1 1 1 0 0 0
    1 1 0 0 0 0 0 0
    1 1 1 1 1 1 0 0
    1 0 1 1 0 1 0 0
    0 0 0 0 0 0 0 0
    1 0 1 0 0 0 0 0
    0 0 0 0 0 0 0 0
```

162	1	0	1	1	1	1	0	0
163	1	0	1	0	0	1	0	0
164	1	0	0	0	0	0	0	0
165	1	0	0	0	0	0	0	0
166	1	0	1	0	0	0	0	0
167	1	0	0	0	0	0	0	0
168	1	0	0	0	0	0	0	0
169	1	0	0	0	0	0	0	0
170	1	0	0	0	0	0	0	0
171	1	0	0	0	0	0	0	0
172	1	0	0	0	0	0	0	0
173	1	1	0	0	0	0	0	0
174	1	1	1	0	1	1	0	0
175	1	1	0	0	0	0	0	0
176	0	0	0	0	0	0	0	1
177	0	1	0	0	0	0	0	1
178	1	0	0	0	0	0	0	0
179	1	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0
181	0	1	0	0	1	0	0	0
182	1	1	1	1	1	1	1	1
183	1	1	0	0	1	1	0	0
184	1	1	0	0	0	0	0	0
185	1	1	0	0	0	0	0	0
186	1	0	1	0	0	0	0	0
187	1	1	1	1	1	1	0	0
188	1	0	0	1	0	1	0	0
189	1	0	0	0	0	1	0	0

190	1	0	1	0	0	0	0	0
191	1	0	1	1	0	1	0	0
192	1	0	1	0	0	0	0	0
193	1	0	0	1	0	0	0	0
194	1	0	0	1	1	1	0	0
195	1	0	0	0	1	0	0	0
196	1	0	0	1	0	0	0	0
197	1	0	0	0	0	0	0	0
198	1	0	0	0	0	0	0	0
199	1	0	0	0	0	0	0	0
200	1	0	0	0	0	0	0	0
201	1	1	0	0	0	0	0	0
202	1	0	1	0	0	0	0	0
203	0	0	0	0	0	0	0	1
204	1	1	1	1	1	1	1	1
205	1	1	1	1	1	1	0	1
206	1	0	0	0	0	0	0	0
207	1	0	0	0	0	0	0	0
208	1	1	0	1	1	1	1	1
209	0	1	0	1	0	0	0	1
210	1	1	0	0	0	0	0	0
211	1	0	0	0	0	0	0	0
212	1	1	1	1	1	0	0	0
213	1	1	1	1	1	1	0	0
214	1	1	1	1	1	1	0	0
215	1	0	1	1	0	1	0	0
216	1	0	0	0	0	0	0	0
217	1	0	0	0	0	0	0	0

```
     1 0 0 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 0 1 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 1 0 0 0 0 0 0
     1 0 0 0 0 0 0 0
     1 1 0 0 0 0 0 0
     1 0 0 0 1 1 0 0
     1 1 1 1 1 1 1 1
     1 1 0 0 0 0 0 0
     1 1 0 0 0 0 0 0
     1 1 0 0 0 0 0 0
     1 1 0 0 0 0 0 0
     1 0 1 1 1 1 0 0
     1 0 1 1 1 1 0 0
     1 0 1 1 1 1 0 0
     1 0 1 1 1 1 0 0
     1 0 1 1 1 1 0 0
     1 0 1 1 1 1 0  0];

% Number of people and number of jobs
[numppl, numjob] = size(G);

% Number of days
numday = 7 * numweek;
```

```

%--------------------------------------------------------------------------

% Objective function
p = max(C);
f = zeros(numday, numflt, numjob, numppl);
for i = 1:1:numppl
    for j = 1:1:numjob
        if G(i, j) == 1
            f(:,:,j,i) = -min(R(j),C(i));
            if ~strcmp(S{i,3},'')
                m = find(strcmp(S{i,3},F));
                for n = 1:1:min(size(Q,2),numweek)
                    f(((n-1)*7+(1:7)),Q(m,n),j, i) = ...
                                                f(((n-1)*7+(1:7)),Q(m,n),j,
                                                    i) - p;
                end
            end
        end
    end
end

%--------------------------------------------------------------------------

% Ensure no one works more than one shift per day
A = zeros(numday, numflt, numjob, numppl, numday, numflt,
    numppl);
for k = 1:1:numppl
```

```
    for i = 1:1:(numday - 1)
        A( i, :, :, k, i, 1, k) = 1;
        for j = 2:1:numflt
            A(i, (j:numflt), :, k, i, j, k) = 1;
            A(i + 1, (1:(j - 1)), :, k, i, j, k) = 1;
        end
    end
    A(numday, :, :, k, numday, 1, k) = 1;
end
Aineq = sparse(reshape(A, numday * numflt * numjob * numppl, ...
                                                            numday * numflt * numppl)');
bineq = ones(size(Aineq,1),1);

%--------------------------------------------------------------------------

% Ensure no one gets more than 7 - numon days off at a time
A = zeros(numday, numflt, numjob, numppl, (numday - (7 - numon)), numppl);
b = -ones((numday - (7 - numon)) * numppl, 1);
for i = 1:1:numppl
    for j = 1:1:(numday - (7 - numon))
        A((j:(j + (7 - numon))), :, :, i, j, i) = -1;
    end
end
Aineq = sparse([Aineq; reshape(A, numday * numflt * numjob * numppl, ...
```

```
                                                (numday - (7 - numon)) * numppl)']);
bineq = sparse([bineq; b]);

%--------------------------------------------------------------------------

% Ensure no one works more than numon days at a time
A = zeros(numday, numflt, numjob, numppl, (numday - numon), numppl);
b = numon * ones((numday - numon) * numppl, 1);
for i = 1:1:numppl
    for j = 1:1:(numday - numon)
        A((j:(j + numon)), :, :, i, j, i) = 1;
    end
end
Aineq = sparse([Aineq; reshape(A, numday * numflt * numjob * numppl, ...
                                                (numday - numon) * numppl)']);
bineq = sparse([bineq; b]);

%--------------------------------------------------------------------------

% No single day off
A = zeros(numday, numflt, numjob, numppl, (numday - 2), numppl);
b = ones((numday - 2) * numppl, 1);
for i = 1:1:numppl
    for j = 2:1:(numday - 1)
```

```
        A(j,:,:,i,(j-1),i) = -1;
        A((j-1),:,:,i,(j-1),i) = 1;
        A((j+1),:,:,i,(j-1),i) = 1;
    end
end
Aineq = sparse([Aineq; reshape(A, numday * numflt * numjob * numppl, ...
                                                    (numday - 2) * numppl)']);
bineq = sparse([bineq; b]);

%------------------------------------------------------------------------------

% No single day on
A = zeros(numday, numflt, numjob, numppl, (numday - 2), numppl);
b = zeros((numday - 2) * numppl, 1);
for i = 1:1:numppl
    for j = 2:1:(numday - 1)
        A(j,:,:,i,(j-1),i) = 1;
        A((j-1),:,:,i,(j-1),i) = -1;
        A((j+1),:,:,i,(j-1),i) = -1;
    end
end
Aineq = sparse([Aineq; reshape(A, numday * numflt * numjob * numppl, ...
                                                    (numday - 2) * numppl)']);
bineq = sparse([bineq; b]);
```

```

%--------------------------------------------------------------------------

% Enforce the on/off pattern

D = logical(triu(ones(7),0)+tril(ones(7),-1*numon)-triu(ones(7),numon));
B = false(7);
p = 0;
for i = 1:1:6
    for j = (i + 1):1:7
        B(i, j) = min(D(:, i) | D(:, j));
        B(j, i) = B(i, j);
        p = p + B(i, j);
    end
end
A = zeros(numday, numflt, numjob, numppl, p, numweek, numppl);
b = -ones(p * numweek * numppl, 1);
for i = 1:1:numppl
    for j = 1:1:numweek
        p = 1;
        for m = 1:1:6
            for n = (m + 1):1:7
                if B(m,n)
                    A(((j - 1) * 7 + (1:7)),:,:,i,p,j,i) = -1;
                    p = p + 1;
                end
            end
```

```
            end
        end
end
Aineq = sparse([Aineq; reshape(A, numday * numflt * numjob * numppl, ...
                                                (p - 1) * numweek * numppl)']);
bineq = sparse([bineq; b]);

%--------------------------------------------------------------------------

% Ensure there's enough people assigned to each job
p = 1;
b = zeros(numday * numflt * numjob, 1);
A = zeros(numday, numflt, numjob, numppl, numday, numflt, numjob);
for k = 1:1:numjob
    for j = 1:1:numflt
        for i = 1:1:numday
            A( i, j, k, :, i, j, k) = -G(:, k);
            b(p) = -N(k);
            p = p + 1;
        end
    end
end
Aineq = sparse([Aineq;reshape(A, numday * numflt * numjob * numppl, ...
```

```
                                                        numday * numflt *
                                                            numjob)']);
bineq = sparse([bineq;b]);

%-----------------------------------------------------------------------------
% Ensure everyone works numon days per week

A = zeros(numday, numflt, numjob, numppl, numweek, numppl);
b = numon*ones(numweek * numppl, 1);
for i = 1:1:numppl
    for j = 1:1:numweek
        A(((j - 1) * 7 + (1:7)), :, :, i, j, i) = 1;
    end
end
A = sparse(reshape(A, numday * numflt * numjob * numppl, numweek
    * numppl)');
b = sparse(b);

%-----------------------------------------------------------------------------

% Find variables with no value
f = reshape(f,1,[]);
Z = f ~= 0;
f = f(Z);
Aineq = Aineq(:, Z);
A = A(:, Z);
Y = sum(Aineq, 2) ~= 0;
Aineq = Aineq(Y,:);
```

```
bineq = bineq(Y);
Y = sum(A, 2) ~= 0;
A = A(Y,:);
b = b(Y);

%------------------------------------------------------------------------------

ub = ones(size(Aineq,2),1);         % Variable upper bounds
lb = ub - 1;                        % Variable lower bounds
intcon = 1:max(size(ub));
ctype = repmat('B',1,max(size(ub)));

% Solve the Binary Integer Linear Program
%[Y,~, exitflag] = intlinprog(f, intcon, Aineq, bineq, A, b, lb, ub);
 [Y,~,exitflag] = cplexmilp(f,Aineq,bineq,A, b,[],[],[],lb,ub,ctype);

if exitflag > 0
    X = zeros(1, numday * numflt * numjob * numppl);
    X(Z) = Y;
    clear Y;
    X = logical(round(reshape(X,numday, numflt, numjob, numppl)));
else
    disp('The schedule is infeasible.');
    return;
end
```

```matlab

%--------------------------------------------------------------------------

W = cell(numppl+1,numweek*7+1);
W{1,1} = 'Personnel';
for j = 2:1:(numweek*7+1)
    W{1,j} = strcat('Day_',num2str(j-1));
end
for i = 1:1:numppl
    W{i+1,1} = strcat(S{i,1},32,S{i,2},32,'(',S{i,3},')');
     for j = 1:1:(7 * numweek)
         for m = 1:1:numflt
             for n = 1:1:numjob
                 if X(j,m,n,i)
                     W{i+1,j+1} = strcat(W{i+1,j+1},10,'Job:',32,
                         ...
                         num2str(n),10,'Shift:',32,num2str(m));
                 end
             end
         end
    end
end

W = cell2table(W(2:end,:),'VariableNames',W(1,1:end));
 writetable(W,'schedule.xlsx');

toc;
```

Bibliography

1. Air Force, 557th Weather Wing . “28th Operational Weather Squadron”. Website. URL: `https://http://www.557weatherwing.af.mil`. Accessed on 2018-05-03.
2. Salem M Al-Yakoob and Hanif D Sherali. “Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations”. *Annals of Operations Research*, 155(1):119–142, 2007.
3. Jonathan F Bard and Hadi W Purnomo. “Hospital-wide reactive scheduling of nurses with preference considerations”. *IIE Transactions*, 37(7):589–608, 2005.
4. Hocine Bouarab, Sophie Champalle, and Martine Dagenais. “Nurse scheduling: From theoretical modeling to practical resolution”. *Resource Optimization in Healthcare*, 2010.
5. Todd E Combs. “A combined adaptive tabu search and set partitioning approach for the crew scheduling problem with an air tanker crew application”. Technical report, Air Force Institute of Technology, Wright-Patterson AFB. OH. School of Engineering and Management, 2002.
6. Patrick De Causmaecker, Peter Demeester, G Vanden Berghe, and Bart Verbeke. “Analysis of real-world personnel scheduling problems”. In *Proceedings of the 5th international conference on practice and theory of automated timetabling, Pittsburgh*, pages 183–197, 2004.
7. Ugur Erdemir. “Optimizing Flight Schedules By an Automated Decision Support System”. Technical report, Air Force Institute of Technology, Wright-Patterson AFB. OH. School of Engineering and Management, 2014.

8. Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. "Staff scheduling and rostering: A review of applications, methods and models". *European journal of operational research*, 153(1):3–27, 2004.

9. Fuat Evren. "A Knowledge-Based Approach to Resource Scheduling in an F-16 Fighter Training Unit.". Technical report, Term Project, Middle East Technical University, 1999.

10. Richard Freling, Dennis Huisman, and Albert PM Wagelmans. "Models and algorithms for integration of vehicle and crew scheduling". *Journal of Scheduling*, 6(1):63–85, 2003.

11. Fred Glover. "Tabu Search: A Tutorial". *Interfaces*, 20(4):74–94, 1990.

12. Ravinder Nanda and Jim Browne. *"Introduction to Employee Scheduling"*. Van Nostrand Reinhold, 1992.

13. Michael L Pinedo. *"Scheduling: Theory, Algorithms, and Systems"*. Springer, 2016.

14. Springer Verlag GmbH, European Mathematical Society. "Encyclopedia of Mathematics". Website. URL: `https://www.encyclopediaofmath.org/`. Accessed on 2018-05-03.

15. Wikipedia Contributors. Plagiarism — Wikipedia, "The Free Encyclopedia". Website. URL: `https://en.wikipedia.org/`. Accessed on 2018-05-03.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 22-09-2018	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From - To)* Sept 2016 - Sept 2018

4. TITLE AND SUBTITLE

OPERATIONAL SQUADRON SCHEDULING

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Khaled S. Almannaei, Major, B.H. Air Force

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-18-S-029

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

28th Operational Weather Squadron (ACC)
905 Patrol Road
Shaw AFB, SC 29152

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The 28th Operational Weather Squadron (28th OWS) is responsible for producing and disseminating mission planning and execution weather analyses and forecasts. The squadron must prepare schedules that meet the needs of their mission while dealing with real-world constraints such as time windows, task priorities, and intermittent recurring missions. The 28th OWS's manning consists of active duty, deployed in-place, reserve, civilian and contract personnel. In this research, a scheduling model and algorithm are provided as an approach to crew scheduling for the 28th Operational Weather Squadron. Scheduling in the 28th OWS is complex and can be time consuming. This model will reduce the time and burden of scheduling the squadron.

15. SUBJECT TERMS

Scheduling, Optimization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Lt Col A. Geyer, AFIT/ENC
U	U	U	U	71	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-6565 x4584; Andrew.Geyer@afit.edu