A DECISION SUPPORT SYSTEM FOR EFFECTIVE SCHEDULING IN AN F-16 PILOT TRAINING SQUADRON

THESIS

Davut Aslan, Captain, TUAF

AFIT/GOR/ENS/03-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GOR/ENS/03-01

A DECISION SUPPORT SYSTEM FOR EFFECTIVE SCHEDULING IN AN F-16 PILOT TRAINING SQUADRON

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

Davut Aslan, MS

Captain, TUAF

March 26, 2003

AFIT/GOR/ENS/03-01

A DECISION SUPPORT SYSTEM FOR EFFECTIVE SCHEDULING IN AN F-16 PILOT TRAINING SQUADRON

Davut Aslan, MS
Captain, TUAF

Approved:

______________________________	26 March 2003
Victor D. Wiley, Maj., USAF (Advisor) Assistant Professor of Operations Research Department of Operational Sciences Air Force Institute of Technology	date
______________________________	26 March 2003
Richard F. Deckro, DBA (Reader) Professor of Operational Research Department of Operational Sciences Air Force Institute of Technology	date

Acknowledgments

I would like to thank several people whose help and support have helped me to make this thesis a reality. First, I would like to thank my advisor, Maj. Victor D. Wiley and reader Dr. Richard F. Deckro for helping me while researching the problem.

Most importantly, I would like my wife and my daughter for their love, support, and understanding during this effort. I most certainly owe them much time and attention for long hours we spent apart while I was researching the problem.

Davut Aslan

Table of Contents

Page

List of Figures

List of Tables

AFIT/GOR/ENS/03-01

Abstract

Scheduling of flights for a flight training squadron involves the coordination of time and resources in a dynamic environment. The generation of a daily flight schedule (DFS) requires the proper coordination of resources within established time windows. This research provides a decision support tool to assist in the generation of the DFS. Three different priority rules are investigated for determining an initial ordering of flights and a shifting bottleneck heuristic is used to establish a candidate DFS. A user interface allows a scheduler to interact with the decision support tool during the DFS generation process. Furthermore, the decision support tool provides the capability to produce a weekly schedule for short-term planning purposes as well as the entire flight training program schedule for long-term planning purposes.

A DECISION SUPPORT SYSTEM
FOR EFFECTIVE SCHEDULING
IN AN F-16 PILOT TRAINING SQUADRON

CHAPTER 1. INTRODUCTION

1.1 Background

Even though the Turkish Air Force (TUAF) flies one of the most advanced aircraft in the world, the F-16, TUAF manually schedules its flights. Akinci AFB is one of the largest bases in Turkey. The 143rd Oncel Squadron (Oncel) at Akinci provides F-16 follow on flight training for newly graduated pilots from undergraduate pilot training and refresher training for pilots who are either changing aircraft types or refreshing their F-16 training. Oncel's primary training workload involves new pilots taking the basic course, known as the B course. In general, the 143rd is the only squadron which offers the B course. Because the F-16 pilot graduates from Oncel will be on active duty for at least the next 15 years, the squadron receives special attention from the TUAF Headquarters.

Maintaining the quality of flight training, as well as keeping up with the training timeline, is the main concern of the TUAF training division. The B course is a 62-sortie flight training program that consists of 3 different flying phases. The first phase of the B course is called Basic Training, which is comprised of two sub phases, basic training (BTR) and basic instrument (BIF). The Air-to-Air (AA) phase is the second phase of the

B course and consists of basic fighter maneuver (BFM), air combat maneuvering (ACM) and basic intercept (BI) sub phases. The third and final phase of the B course is Air to Ground (AG) phase. The AG phase is comprised of 6 sub phases. Upon completion of the AG phase, Instrument (IQ) and Tactical qualification (TQ) check rides must be accomplished in order to successfully graduate from the B course. Additionally, academic and device training occur throughout the B course, complementing and supporting flight training. Based on a class of 20 Student Pilots (SPs), Oncel is given 140 weekdays to graduate pilots for TUAF (35 weekdays for academics, simulator and device training and 105 weekdays for flight training).

Each year approximately 50-student pilots go through the B course training in Oncel. The squadron, depending on the phase of the flight, flies approximately 20 missions and more than 40 sorties each day with most missions requiring more than one aircraft. The pilots in the squadron are classified in two groups according to their instructor status. The first group contains all qualified pilots also known as Bandits. The second group is a subset of the Bandits who are qualified to be flight instructors, the Instructor Pilots (IPs). Whenever a mission requires more than one aircraft, then it also needs additional pilots. These pilots fly the support sorties for that mission and can be either IPs or Bandits. This requirement for additional pilots, either current IPs or Bandits, complicates the scheduling procedure and makes it more difficult to de-conflict with the other training squadrons.

In addition to student sorties, Oncel sometimes must schedule operational sorties (Oncel is still an operational squadron) such as DACT-AAR (combat training with a different type of aircraft followed by air refueling) or flights for the Instructors and

Bandits to maintain their currency for the various missions. To accomplish all of these tasks Oncel must have a higher mission throughput rate compared to other operational squadrons.

Currently, one of the IPs at Oncel acts as the Program Officer and two, possibly three, IPs act as the schedulers. The Program Officer receives his assignment from headquarter, but the Squadron Commander assigns the schedulers. The Program Officer is the highest-ranking officer in the training division of the squadron. He supervises the schedulers and makes the last check on the schedule before he signs it. Schedulers rotate annually dependent on the Commander's decision. When schedulers change, the Squadron Commander keeps at least one of the current schedulers to maintain the corporate knowledge gained from experience. During these transitions, less efficient schedules may be generated while the new scheduler gains experience. On weekdays, schedules comprising the next-day's flights are adjusted for changes due to the maintenance, weather, and sickness. At the end of the week, a tentative 5-day schedule is established for the upcoming week.

Generating the daily schedule requires a great deal of the scheduler's time. The amount of time spent making the schedule mainly depends on the current flying phase and the experience of the scheduler. *Any* improvement in the scheduling process will reduce the workload of the scheduler and can potentially provide better schedules with higher mission rates. Reducing the time required to produce the schedule allows more time for the SPs to prepare for the missions they will fly the next day. It also gives IPs additional time to review a SP's records concerning previous flights. This will allow the IPs to improve training quality and flight safety. Finally, a reduction in time spent scheduling will allow more time for the program and scheduling officers to focus on their primary duties as IPs.

In addition to the benefits it provides to Oncel squadron, better schedules generated more rapidly also gives more time for maintenance to allocate the aircraft for the scheduled missions. Maintenance enters the tail numbers and the locations of the aircraft next to the missions after the draft schedule is generated. Maintenance, with an earlier schedule generation, will also have more time to change the configuration of the aircraft if necessary without working past the close of business (COB).

Similarly, generating the weekly schedule also consumes a great deal of the scheduler's time. Even though it is a tentative schedule, it still takes a half-flying day for the scheduler to complete the 5-day schedule sheet. For the 5-day schedule, the scheduler has to generate approximately 20 missions each day composed of both training and operational flights. The scheduler must allocate enough personnel for classrooms, simulators, and flight training as well as ensure that enough aircraft are available. Therefore, improvements in the weekly scheduling process will also reduce the workload of the scheduler, once again allowing more time for the program and scheduling officers to focus on their primary duties as IPs.

The squadron scheduler determines when a SP flies a mission or simulator and with whom, when a SP attends a class or physical training and when an IP flies a mission or performs other duties such as range officer or simulator instruction. An improved schedule, therefore, provides the benefit of better training opportunities for the SPs and a more balanced activity list for the other members of the squadron.

The current scheduling in Oncel is *ad hoc* with no automated ability during the scheduling and rescheduling processes. A distinction is made between Scheduling and Rescheduling. Scheduling refers to the establishing of the schedule before the start of the day based on the previous day's activities. Rescheduling refers to adjusting the planned schedule by reassigning new missions, new SPs, new IPs or bandits, new aircraft, new operational areas during the execution of the Daily Schedule (i.e. during the flying day). (The weekly schedule does not require rescheduling).

A scheduler starts making the next day's schedule after all of the current day's mission de-briefings are finished. The next day's schedule accounts for "effective" and "non-effective" missions. An effective mission allows SPs to continue their normal program while a "non-effective" mission requires SPs to reaccomplish a mission before proceeding to the next level of training.

Currently, the scheduler starts with a large paper chart designed for daily scheduling or a spreadsheet designed in the same manner. The scheduling consists of several unwritten steps executed by all the schedulers. The squadron scheduler firsts coordinates with maintenance for the available F-16D and F-16C aircraft and their configurations (each mission requires a distinct configuration). Sometimes configurations must be altered or spare aircraft with the required configurations for the "must-fly" student missions must be prepared. Next he determines how many IPs, SPs, Bandits and reserve pilots are available (reserve pilots are drawn from the other non-flying units; they can be IP or Bandit).

The next step determines how many flight areas are assigned to Oncel and, if necessary, coordinates with the other squadrons (there are seven areas available for

Akinci AFB) to deconflict the flight area air spaces. Because Akinci AFB has three squadrons, each needs different take off and landing time slots. Therefore the scheduler has to make sure that he uses the assigned time slots for Oncel. If necessary, he can further coordinate with the other squadrons to switch areas or time slots.

After completing all these steps and scheduling operational flights, he can start making the training schedule. In addition to the flight schedule, he also has to schedule the simulator(s), academic courses, and assign persons to their daily duties. Furthermore, he must account for, on an individual basis, each activity performed by any pilot during the day.

1.2 Problem Statement

Oncel's schedule encounters more dynamic changes than any other flight squadron's schedule within the TUAF. Weather is one of the main factors that may affect flight scheduling. Aircraft breakdowns or SP "non-effective" missions are some other factors that may require changes to the daily schedule. This re-scheduling is generally done by the duty scheduler, or other schedulers at a moment's notice to try and save the mission.

Any tool that can assist scheduling and rescheduling will decrease the amount of time spent on these processes, increase the efficiency of the utilization of resources and also prevent overtasking of individuals. It will also enable any scheduler to make necessary changes easily when the duty scheduler is not available. Additional benefits include more time for support (maintenance and other flight divisions) to prepare.

1.3 Scope

The scope of this thesis research is limited to planning and generating of a daily flight schedule (DFS), which consists of operational and training missions while maximizing the resource use and maintaining balance in the squadron. In addition to the DFS, this research produces a tentative weekly schedule, which consists of training missions for the next week. The model developed in this research also provides rescheduling ability to schedulers to maximize the SP activities while maintaining the quality of the training, and meeting training requirements, and ensuring course completion within the allotted time.

1.4 Methodologies

A fighter training squadron scheduling support tool (FTSSST) is developed to assist Oncel squadron scheduling division. The FTSSST is a spreadsheet based scheduling software and has been developed according to the squadron scheduler's familiarity with the current spreadsheet or chart method approach. A database is created to store and update all the data necessary for DFS and weekly schedule. Adding a scheduling engine and making the software user friendly are the main modifications made to the spreadsheet. Every SP's flown flight sorties or simulator sorties are also stored by FTSSST on individual basis at the end of each flight day for tracking purposes. In addition to storing these sorties, the evaluation of the each activity performed by a SP is also stored to provide a foundation for the next day's DFS. The sorties flown by the IPs and the Bandits are recorded in addition to the SP activities at the end of the flight day. The duties performed by the IPs and Bandits are maintained and used to balance additional duty assignments.

The user-friendly buttons, coupled with other Graphical User Interfaces (GUI) such as coloring and tables, provides ease of use to the scheduler. The code that runs the scheduling engine and other features exists behind the GUIs; pushing a button activates the codes that perform the associated scheduling functions (i.e. mission order, duty assignments etc.)

The FTSSST allows schedulers to over-ride and modify the DFS before, during or after the DFS is generated. Using his experience and insights, the squadron scheduler can change the order of the mission list or the first take off time before the schedule is generated. He can alter the names of the pilots who are assigned for duties during the generation phase. The FTSSST, providing support in generating schedules, also has the flexibility to incorporate the judgment and the experience of the schedulers and commanders.

Additionally, the FTSSST assigns C and D model F-16 aircraft with tail numbers and their locations to the DFS with correct configuration appropriate to each mission besides generating the DFS. This aircraft assignment task reduces the workload for maintenance and stops unnecessary configuration changes, which lessens the wear on aircraft and accessories.

At the end of each flight day the realized, flown schedule is stored to a separate spreadsheet for recording purposes.

1.5 Summary

This chapter began by providing a background on the scheduling process of Oncel. The structure of the scheduling division was then introduced. A distinction between the Scheduling and Re-scheduling was made and the problem was addressed.

Chapter 2 presents a review of literature and background to the scheduling problem. Chapter 3 details the development of the scheduling and a new Heuristics used to solve the Scheduling problems of Oncel squadron. Chapter 4 presents the results of the scheduling rules defined in chapter 3 after they are used to solve a real problem. Chapter 5 gives the summary of the research, contributions and recommendations for the future research.

CHAPTER 2. LITERATURE REVIEW

General

To lay a foundation for this research, this chapter covers the topics related to scheduling theory. In addition to a review of the pertinent literature on flight scheduling, the chapter also provides the background on the scheduling environment of 143rd Oncel Squadron. The scheduling portion of this research uses a new construction heuristic. Therefore, a description of the heuristic is also presented. The search methods employed to generate an initial feasible solution to the resulting Mixed Integer Programming (MIP), Multi-Modal Resource Constrained Project Scheduling Problem (MMGRCPSP) heuristic is reviewed next. To improve upon this initial solution, a new initial mission order heuristic is developed. In addition to this order heuristic a shifting bottleneck heuristic (SBH) is developed to account for scarce resources thorough out the timeline. Therefore, a section is devoted to heuristics and heuristic methods. The implementation of the mission order heuristics and flight scheduling shifting bottleneck heuristic (FSSBH) uses Visual Basic for Applications (VBA). A section on object oriented programming (OOP) rounds out the chapter.

2.1 Scheduling Theory

This section introduces concepts from scheduling theory and topics related to project management (PM) and scheduling process. "Scheduling deals with the allocation of limited resources to tasks over time. It is a decision-making process with the goal of optimizing one or more objectives" [Pinedo, p.1, 2002]. The scheduling process exists as a decision-making process in nearly all operational environments. The scheduling function may also face a variety of different problems in a service organization, e.g., it can be dealing with the reservation of resources [Pinedo, p.6, 2002] (e.g., the assignment of aircraft to a future mission even though they are not currently initialized).

> The resources and tasks in an organization can take many forms. The resources may be machines in a workshop, runways at an airport, crews at a construction site, a processing unit in a computer environment, and so on. The tasks may be operations in a production process, take offs and landings at an airport, stages in a construction project, executions of computer programs and so on. [Pinedo, p.1, 2002]

"The objectives can also take many forms. One objective may be the minimization of the completion time the last task (makespan), and another may be the minimization of the number of tasks completed after their respective due dates." [Pinedo, p.1, 2002].

"Scheduling has attracted much interest from the academia. Most theoretical research on this topic is geared towards simple machine scheduling problems." [Evren, p.4, 1999]. The scheduling process in the operational world is more complex than the regular theoretical machine scheduling models. Empirically, scheduling problems that are relevant to resource scheduling environments may be summarized as:

- Theoretical models usually assume that there are *n* jobs to be scheduled and that after scheduling these *n* jobs the problem is solved. In reality, new jobs are added to the system continuously. The dynamic nature of resource scheduling in service organizations may require that slack times be built into the schedule in expectation of the unexpected.

- Theoretical models usually do not emphasize the resequencing problem. In practice, some random event may require major changes and the rescheduling process may have to satisfy certain constraints. This is sometimes referred to as reactive scheduling. Thus, stochastic scheduling environments such as flight training in inclimate weather conditions might benefit from robust schedules instead of some optimality objective.

- Machine environments in the real world are often more complicated than the ones considered in general scheduling theory.

- In most mathematical models, the weights (priorities) of the jobs are assumed to be fixed, that is, they do not change over time. In practice, the weight of a job often fluctuates over time.

- Mathematical models often do not take preferences into account. A scheduler may favor some assignment for a reason that cannot be incorporated into the model.

- Most theoretical research has focused on models with a single objective. In the real world, there are usually a number of objectives. One such example arises in flight scheduling. Minimizing student sortie re-fly rate (rework in a production environment) objective might conflict with maximum resource utilization associated with minimum makespan objective. Due-date tightness of the jobs might dictate the relative importance of each objective in flight scheduling.

- In practice, flight scheduling is strongly affected by the assignment of shifts and the scheduling of overtime within certain safety and regulatory constraints. Whenever the workload appears to be excessive and due dates appear to be too tight, the decision-maker has the option to schedule overtime or put in extra shifts to meet the committed completion dates. [Evren, p.5, 1999].

"A project is a systematic enterprise designed to accomplish some specific non-routine or low-volume task" [Shtub, *et al*, p.1, 1994]. Project management is the process of planning, scheduling, and overseeing the activities of a project [Calhoun, p.23, 2000]. Every project has constraints. The primary ones are the trade off between time, resources and the performance criteria. The choices to select and balance these constraints have to be made to define the project so that it can be managed. For the purpose of this research, project management, then, is the allocating of the resources, i.e. aircraft, pilots and areas, to a finite set of missions belonging to each SP, which must be scheduled in agreement with certain precedence requirements over a planning horizon. In the context of this thesis the project is the daily flight schedule (DFS).

This section is intended to achieve several objectives. First, it introduces the reader to key concepts from scheduling theory and then lays the foundation for the development of a heuristic to obtain an initial solution to DFS. Drawing analogies between flight scheduling, project scheduling and job shop scheduling accomplishes this. Third, the section reviews the integer linear programming (ILP) model developed by Van Hove to theoretically produce an optimal solution for air combat planning. The model is established to demonstrate the enormity of the problem in terms of variables and constraints, justifying the practical need for a heuristic technique albeit with the loss of guaranteed optimality.

2.1.1 Gannt Charts

A Gannt chart is a horizontal bar chart developed as a production control tool. It is frequently used in project management. A Gannt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project. It is constructed with a horizontal axis representing the total time span of the project and the vertical axis representing the tasks (activities) that make up the project. Gantt charts give a clear illustration of project status. However they do not provide a clear indication of task dependencies.

2.1.2 Single Machine Models

> The single machine problems have been well studied because single machine models are important for various reasons. Many variations of measures of performance, job characteristics, precedence and release times provide a variety of single machine models. [M'Hallah and Bulfin, p.1, 2001].

The single machine environment is a special case of all other environments. For this reason the results obtained for a single machine model provide a basis for heuristics that are applicable to more complicated machine environments which are often decomposed into subproblems that deal with single machines.

In order to completely understand the behavior of a complex system, it is vital to understand the workings of its components. Quite often the single machine problem appears as an elementary component in a larger scheduling problem. For example, a complicated machine environment with a single bottleneck stage can be treated as embedded single machine problem, which may also determine the properties of the entire schedule [Baker, p.10, 1974]. This single machine analysis and its results can then be incorporated into a complex system.

A basic single machine problem is characterized by the following conditions: a set of *n* independent jobs available for processing at time zero, with set up times independent of job sequence and included in processing times, job descriptors are known in advance. A machine is continuously available and is never kept idle while work is waiting. Preemption is not allowed. The processing time, ready time and due date of each job are the known in advance. Completion time, Flow time and Lateness are generated as a result of scheduling decisions [Baker, p.11-12, 1974]. In most single-machine environments makespan does not depend on the sequence and therefore it is not important.

2.1.3 Parallel Machine Models

"From a theoretical view point, it is a generalization of the single machine and a special case of the flexible flow shop. From a practical point of view, it is important because the occurrence of the resources in parallel is common in the real world" [Pinedo, p.93, 2002]. When a job *j* arrives in a parallel machine environment, it may be processed on any one of the *m* machines or on any one that belongs to a given subset [Pinedo, p.14, 2002]. Minimizing the makespan (completion time of the last job) is one of the most utilized objectives when dealing with parallel machines. "In practice one often has to deal with the problem of balancing the load on machines in parallel; by minimizing the makespan objective, only the allocation process is important" [Pinedo, p.94, 2002].

A flow shop is an environment where machines are set up in series and the jobs have to follow the same route. In many facilities, every job has to go through a number of operations. These operations often have to be done on all jobs in the same order. A flexible flow shop is a more general environment comprised of a number of stages in series with a number of machines in parallel at each stage. This environment is analogous to the scheduling of a single phase in the flight program.

The Shortest Processing Time (SPT) – Longest Processing Time (LPT) rule is optimal for the $F2 \mid\mid C_{max}$ problem (flow shop with two machines and an objective of minimizing the makespan). Jobs can be partitioned into two sets, with set 1 containing the all the jobs with $p_{1j}<p_{2j}$ (processing time of a job *j* on machine 1 is less than the processing time of the same job *j* on machine 2) and set 2 contains all the jobs with $p_{1j}>p_{2j}$ (processing time of a job *j* on machine 1 is more than the processing time of the same job j on machine 2). The jobs in set 1 go first (SPT) and jobs in set 2 follow (LPT). This schedule is referred to as an SPT (1)-LPT (2) schedule. Unfortunately, the SPT (1)-LPT (2) schedule structure does not always give optimum results for flow shops with more than two machines. However, minimizing the makespan in $Fm \mid prmu \mid C_{max}$ (flow shop with *m* machines where jobs are chosen arbitrarily from the ready list and processed on the machine based on the selected permutation order) environment can be formulated as a Mixed Integer Program (MIP). See Figure 1 for the MIP associated with Fm | prmu | C_{max}.

Variables & Parameters	
X_{jk}	Equals 1 if job j is the kth job in the sequence; 0 otherwise
I_{ik}	Idle time on machine i between the processing of jobs kth and (k+1)th
W_{ik}	Waiting time of the job in the kth position between machines i and (i+1)
P_{ij}	Processing time of job in the kth position on machine i

$$\text{Min} \quad \left(\sum_{i=1}^{m-1}\sum_{j=1}^{n} X_{j1}\, P_{ij} + \sum_{j=1}^{n-1} I_{mj}\right)$$

$$\text{St} \quad \sum_{j=1}^{n} X_{jk} = 1 \qquad k = 1, \ldots\ldots, n,$$

$$\sum_{k=1}^{n} X_{jk} = 1 \qquad j = 1, \ldots\ldots, n,$$

$$I_{ik} + \sum_{j=1}^{n} X_{j,\,k+1} + W_{i,\,k+1} - W_{ik} - \sum_{j=1}^{n} X_{jk}\, P_{i+1,j} - I_{i+1,k} = 0$$

$$k = 1, \ldots\ldots, n\text{-}1 \;\; ; \;\; i = 1, \ldots\ldots\ldots, m\text{-}1,$$

$$W_{i1} = 0 \qquad i = 1, \ldots\ldots\ldots, m\text{-}1,$$

$$I_{1k} = 0 \qquad k = 1, \ldots\ldots, n\text{-}1\,.$$

Figure 1. MIP Formulation [Pinedo, p.137, 2002]

The first set of constraints provides that only one job can be assigned to position *k* for any *k*. The second set of constraints specifies that job *j* has to be assigned to exactly one position. The third set of constraints relates the decision variables X_{jk} to exactly one position [Pinedo, p.137, 2002]. In the DFS, only one mission can be placed the k^{th} position in the mission order list (MOL). There is a delay between the takeoff times and area use as a result of MOL which is analogous to the idleness of the machine waiting for the next job in the order.

If the routes are fixed for the jobs and are the same for each job, then the model is called a job shop. In a flexible job shop, instead of *m* machines, there are *c* work centers with a number of identical machines in parallel at each *c* work center. If job *j* requires processing at any center, any one of the machines can do the processing at the center [Pinedo, p.15, 2002]. This is analogous to scheduling any phase of the flight program in which each mission needs certain types of resources.

> Consider a directed graph G with a set of nodes and two sets of arcs A and B. The nodes N correspond to all the operations (i, j) that must be performed on the n jobs. The so-called conjunctive (solid) arcs A represent the routes of the jobs. If arc (i, j) → (k, j) is part of A, then job j has to be processed on machine i before it is processed on machine k. Two operations that belong to two different jobs and that have to be processed on the same machine are connected to one another by two so-called disjunctive (broken) arcs that go in opposite directions. The disjunctive arcs B form m cliques of double arcs, one clique for each machine. [Pinedo, p.158, 2002].

A feasible schedule is a selection of one disjunctive arc from each pair and the resulting graph is not acyclic (Figure 2).

Scheduling in a parallel machine shop may be considered a two-step process. First, one should determine which jobs should be allocated to which machine. Second, one has to determine the sequence of jobs allocated to each machine [Pinedo, p.94, 2002].

In this research, this sequence is subject to precedence constraints while minimizing the makespan by maximizing throughput of daily sorties.

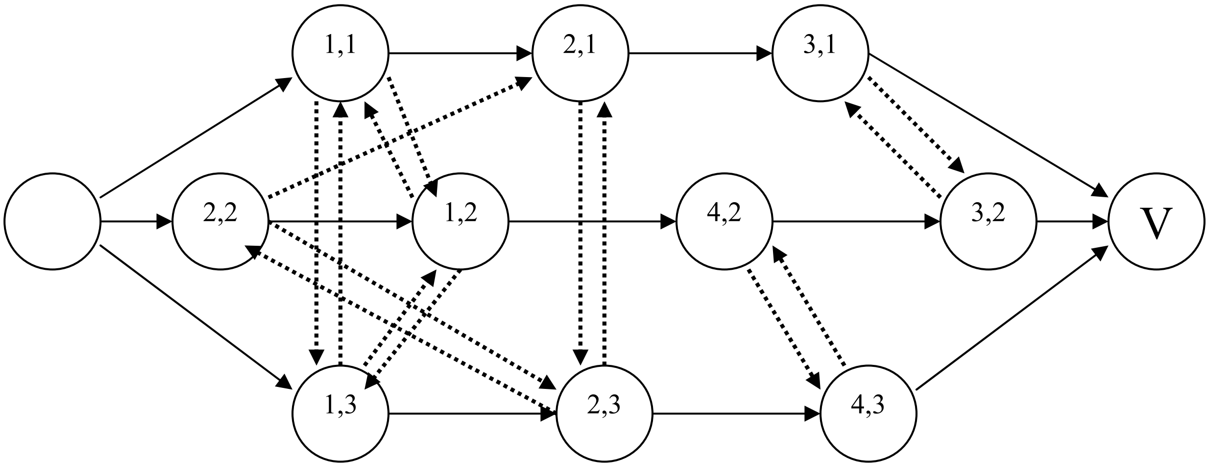


Figure 2. Directed graph for job shop with makespan as objective [Pinedo, p.158, 2002]

There are several mathematical programming formulations for the job shop, but the formulation most often used is the disjunctive programming formulation, which is closely related to the disjunctive graph representation of the job shop. "Minimizing the makespan in a job shop is a very hard problem, and solution procedures are based on either enumeration or heuristics" [Pinedo, p.160, 2002].

If the route of a job is immaterial and if the scheduler can decide in which route the job will go, then the model is referred as an open shop. Longest Alternate Processing Time (LAPT) first rule yields an optimal schedule for $O2 || C_{max}$(Open shop with two machines and objective is to minimize the maximum completion time). However, a more general rule called the Longest Total Remaining Processing on Other Machines first rule is applied to models, but this does not always result in optimal schedule as $Om || C_{max}$ is NP-hard when $m \geq 3$ [Pinedo, p.189, 2002].

2.1.4 Precedence Constraints

"Precedence constraints may appear in a single machine or in a parallel machine environment, requiring that one or more jobs may have to be completed before another job is allowed to start its processing" [Pinedo, p.16, 2002]. There are several types of precedence constraints. The most common type of precedence constraint is the finish-start type. It is used to specify that a predecessor activity must end before its successor activity may start. Other common types are start-finish, start-start, and finish-finish [Calhoun, p.27, 2000].

The amount of precedence constraints among activities in a project may make the project hard to express or formulate mathematically. For this reason graphical representation of precedence constraints are used frequently [Shtub, *et al*, p.321, 1994]. Shtub, *et al* state two ways to represent precedence constraints. One way to represent is by an activity on the arc (AOA) diagram (Figure 3). In AOA representation, a node indicates the end of an activity and the duration of the activity is shown on the arc. If an arc is directed from node *i* to node *j* then *j* can only begin after *i* is completed. The critical path is shown with the boldface arrow in Figure 3. To minimize the makespan of a project in $P\infty \mid prec \mid C_{max}$ (unlimited number of machines in parallel, jobs are processed with precedence relationship, and the objective is to minimize the makespan), Pinedo defines an algorithm that finds the optimal schedule [Pinedo, p.97, 2002]. In this special case where there is an unlimited number of machines, the start of the processing of some jobs usually can be postponed without increasing the makespan. These are referred as the slack jobs. The amount of slack time for job *j* is the difference between its latest possible completion time and its earliest possible completion time. The jobs that cannot be postponed are referred to as the critical jobs. The critical path(s) is comprised of these critical jobs [Pinedo, p.97-98, 2002]. The length of the critical path is the sum of the durations (processing times) of every activity on it.

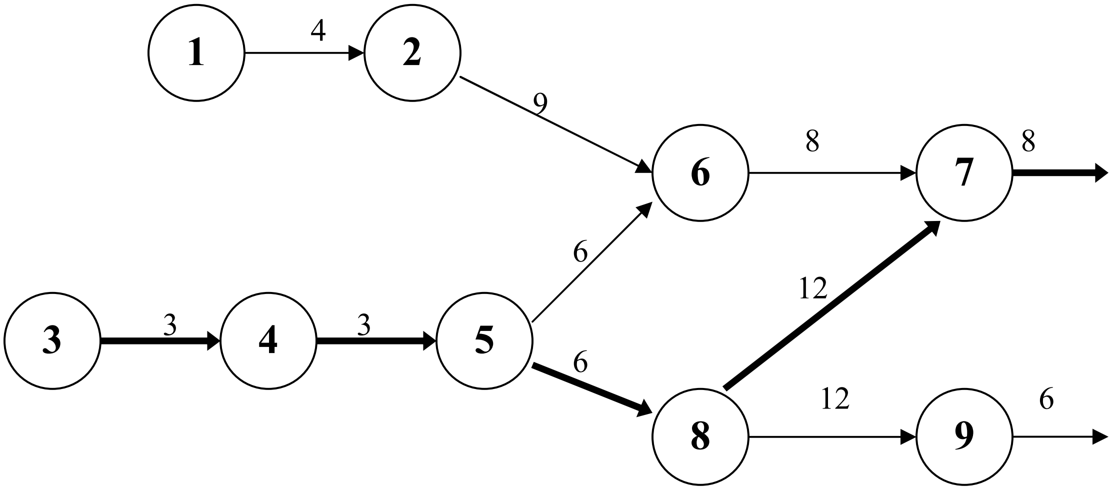


Figure 3. AOA precedence constraint graph with critical path [Pinedo, p.98, 2002]

The other way to represent precedence constraints is by an activity on the node network (AON).

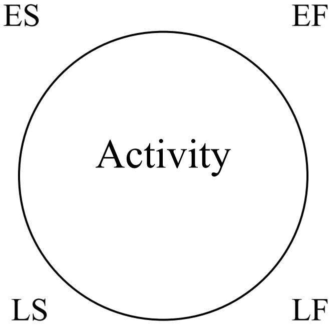


Figure 4. AON representation of an activity

A node in AON (Figure 4) represents an activity in the network and it may display the information about that activity such as duration (processing time), early start (ES), late start (LS), and late finish (LF). Arcs indicate the precedence relationships among the activities.

Contrary to $1 | prec | C_{max}$ and $P\infty | prec | C_{max}$ (single machine and infinite machines, jobs are processed with precedence relationship, and the objective is to minimize the makespan, respectively) problems, the $P_m | prec | C_{max}$ problem with $2 \leq m < n$ is strongly NP-hard. Even the special case with all processing times being proportionate (equal to 1) (i.e., $P_m | p_j=1, prec | C_{max}$) is not easy. On the other hand, constraining the problem further and assuming that the precedence graph takes the form of a tree (either an intree or an outtree) results in a problem (i.e., $P_m | p_j=1, tree | C_{max}$), which is easily solvable [Pinedo, p.99, 2002].

This particular problem leads to a well-known scheduling rule, the Critical Path (CP) rule. The CP rule gives the highest priority to the job at the head of the longest sequence of jobs in the precedence graph (ties may be broken arbitrarily). The single job with no successors is called root and located at level 1 on an intree. The jobs immediately preceding the root are located at level 2 and so on (Figure 5). But in an outtree, all jobs with no successors are located at level 1. Jobs that are immediate predecessors of jobs at level 1 are said to be at level 2 and so on (Figure 5).

It is obvious from these definitions that CP rule is equivalent to the Highest Level first rule. The jobs with no predecessors may be referred as starting jobs [Pinedo, p.99, 2002].

In the case of intrees the CP rule and Largest Number of Successors (LNS) first rule are equivalent. Under the LNS rule the job with the largest number of successors in the precedence constraint graph has the highest priority.

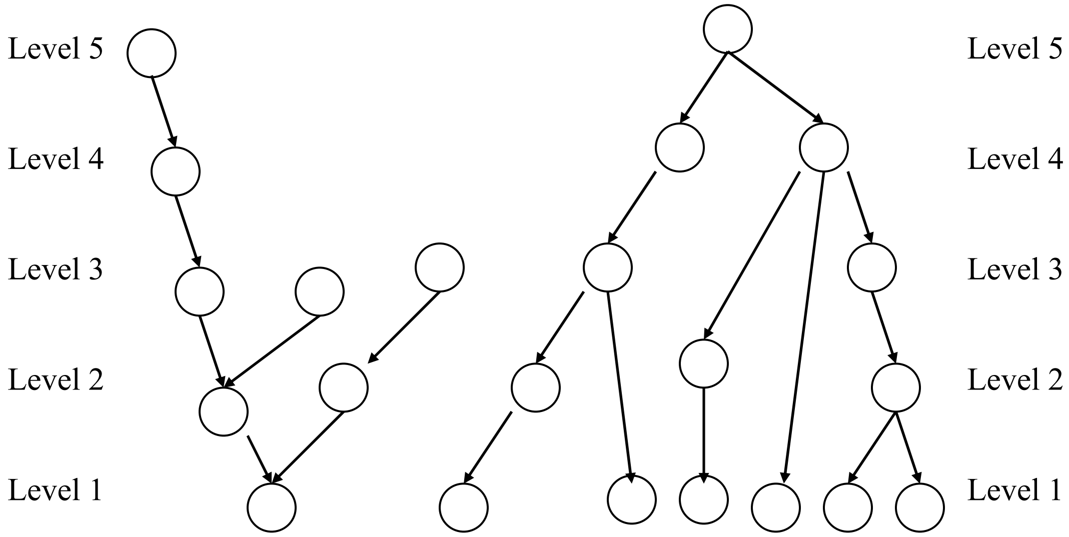


Figure 5. Intree and Outtree

Therefore LNS rule also results in an optimal schedule in the case of intrees (P_m | p_j=1, intree | C_{max}) as well as it gives an optimal schedule for P_m | p_j=1, outtree | C_{max}. The LNS rule is not optimal with arbitrary constraints. [Pinedo, p.102, 2002].

2.2 The Resource Constrained Project Scheduling Problem

A linear programming (LP) model may be used to find the CP for a project scheduling problem (PSP). The simple PSP model assumes that unlimited resources are available to the activities and is modeled with one continuous decision variable for each activity. Van Hove adapted such a model, which is developed to handle limited resources as an expansion of the PSP model (Figure 6). This model is called Resource Constrained Project Scheduling Problem (RCPSP). The objective function (1) minimizes the makespan and equation type (2) states the precedence constraints. Equation type (3) enforces the resource use stay in the limits for each resource available in each period.

The constraints represented by (4) allow each activity to be completed once in one of the possible periods. Equation type (5) forces each decision variable to be binary.

Parameters:

A	the set of all activities
K	the set of all resources
P	the set of all activity precedence pairs
n	the last activity in the network
g	the project deadline
τ_i	the duration of activity i
e_i	the earliest completion time for activity i
l_i	the latest completion time for activity i
r_{ik}	the amount of resource k required by activity when being
R_{jk}	the amount of resource k available in period j

Variables:

x_{it}	1 if activity i finishes in period t; 0 otherwise

$$\text{Minimize} \sum_{t=e_n}^{l_n} t\,x_{1t} - \sum_{t=e_1}^{l_1} t\,x_{1t} \qquad (1)$$

Subject to

$$\sum_{t=e_n}^{l_n} t\,x_{1t} - \sum_{t=e_1}^{l_i} t\,x_{it} \geq \tau_n \quad \forall\ (i,n) \in P \qquad (2)$$

$$\sum_{i \in A} \sum_{t=j}^{j+\tau_i-1} r_{ik}\, x_{it} \leq R_{jk} \quad \forall\ k \in K \text{ and } j=1,\dots,g \qquad (3)$$

$$\sum_{t=e_i}^{l_i} x_{it} = 1 \quad \forall\ i \in A \qquad (4)$$

$$x_{it} \in \{0,1\} \quad \forall\ (i,t) \qquad (5)$$

Figure 6. RCPSP Formulation

The PSP formulation requires only a single continuous decision variable for each activity. The RCPSP makes it necessary to have a series of binary decision variables for activities to account for per period resource consumption. This results in a significant increase in the number of decision variables in the RCPSP formulation in comparison to the PSP formulation.

2.2.1 The Multi-Modal Resource Constrained Project Scheduling Problem

The RCPSP formulation may be extended to the situation where the activities can be processed in one of a number of possible execution modes. The amount and the type

Parameters:
A the set of all activities (for this research related to student m)
K the set of all resources
P the set of all activity precedence pairs
M_i the set of all execution modes for activity i
τ_{im} the duration of activity i in mode m
e_{im} the earliest completion time for activity i in mode m
l_{im} the latest completion time for activity i in mode m
r_{imk} the amount of resource k required by activity when being executed in mode m
R_k the per period availability of resource k
Variables:
x_{imt} 1 if activity i starts in period t and is executed in mode m

$$\text{Minimize} \sum_{m\in M_n} \sum_{t=e_n}^{l_n} t\, x_{nmt} \qquad (6)$$

Subject to

$$\sum_{n\in M_j}\sum_{t=e_j}^{l_j} (t-\tau_{jm})\, x_{jnt} - \sum_{n\in M_i}\sum_{t=ei}^{l_i} t\, x_{im't} \geq 0 \qquad \forall\, (i,j) \in P \qquad (7)$$

$$\sum_{i\in A} \sum_{m\in M_i}^{l_j} \sum_{t=j}^{j+\tau_i-1} r_{imk}\, x_{imt} \leq R_k \qquad \forall\, (j,k) \in K \qquad (8)$$

$$\sum_{m\in M_i} \sum_{t=e_{im}}^{l_{im}} x_{imt} = 1 \qquad \forall\, i \in A \qquad (9)$$

$$x_{imt} \in \{0,1\} \quad \forall\, (i,m,t) \qquad (10)$$

Figure 7. Van Hove's adaptation of Boctor's MMRCPSP Formulation

of the resources consumed depend on the mode selected for the activity. Van Hove [p.19, 1998] adapted Boctor's multi-modal model [Boctor, p.350, 1996].

The objective function (6) of the model is again minimizing the makespan. The major difference between this model and RCPSP is the decision variable.

2.2.2 Generalized MMRCPSP

In the previous formulation, the precedence constraints were strictly start-to-end. These type of constraints are not flexible enough to model mission sequencing in DFS. For example, if activity *j* follows *i* then start-to-end constraints force mission *j* to wait until all aircraft employed in mission *i* have landed and their turn time has expired. In DFS mission j cannot be scheduled in the same day if there is a precedence relation with mission *i* but if the mission *j* only follows the mission *i* in the MOL then it can be scheduled in the DFS with enough takeoff time and other resource usage time de-conflictions. Hence this type of precedence constraint is insufficient for the DFS. Generalized precedence constraints may be used to impose any timing requirement mandatory for an operational scenario. MMGRCPSP is the abbreviation for the generalized MMRCPSP and it will be described later in this chapter.

2.2.3 Doubly Constrained Resources

A clean connection between the scheduling theory and that of DFS is required. These associations are provided in table 1.

In the DFS, the number of allocated aircraft assigned to Oncel on a particular day equals the limit on how many aircraft may be tasked during the same time window. However, this is not the limit on how many aircraft may be tasked throughout the day.

Table 1. Terminology Associations

Scheduling	Daily Flight Schedule (DFS)
Activities or Jobs	Missions to be flown each day by each student
Mode	IP, Bandit, Area, Aircraft, Simulator selection, a combination vector of these resources consumed in a mode
Processing time	Mission duration
Precedence Constraint	Mission timing requirement dependent on the resources and SP activity list. In addition, SP must complete mission i before mission j.
Resource	IP, Bandit, Area, Aircraft, Simulator, CFT, AFT

In general, each aircraft can fly more than one mission in a flight day –provided that the crews are available, no malfunctions occurred and there is sufficient time to turn an aircraft. The number of sorties an individual aircraft can fly per day is associated with its turn rate. The number of sorties Oncel may fly per day is equal to the turn rate multiplied by the number of allocated aircraft in the unit.

Let K be the set of units in the problem and $k \in$ K. The renewable aspect of k is the limit R_k, the actual number of allocated aircraft assigned to Oncel. Now let tr_k be the turn rate for unit k. The nonrenewable aspect of the resource k is the limit N_k which is given by $R_k \bullet tr_k$. [Van Hove, p.43, 1998].

Each resource in the flight schedule is a doubly constrained resource. In other words, the assets associated with the resource are renewable and nonrenewable.

2.2.4 Multi Modal Activities

A mission order list (MOL) determines the activities for a DFS. The MOL lists the current flyable missions for Oncel along with the resource use modes for each activity. The information associated with an option includes the number and type of aircraft, pilots, areas, and simulators required. A valid execution mode(s) for an activity is given by including the number and types of resources necessary to complete the activity. The mission duration is a fixed time and is independent from the execution mode(s). However, a mission can use different level of each resource based on the mode selected. For example, if a mission requires two aircraft, then it may be flown with two F-16D models or 1 F-16D and 1 F-16C or 2 F-16C models. In addition to the aircraft selection decision, it may be flown with 2 IPs or 1 IP and 1 Bandit. When we combine these two resources in modes for this mission, this mission can be flown in 6 different modes (3 x 2).

For this research, an activity consists of the following sub activities listed in the order they are performed: Briefing, ground operations, ingress, area work, egress, landing, engine shutdown and de-briefing. The processing time of each sub activity is fixed except for area work, which is mission specific. The process time of an activity is the sum of process times of sub activities. Since the process time of area work differs for each mission, the duration of the activities are different from each other.

2.2.5 MMGRCPSP Formulation

In the formulation of Van Hove's MMGRCPSP problem the objective function again minimizes the makespan of the schedule. For this research, although minimizing the makespan is an important objective, overall maximizing daily sorties produced is the main objective of the squadron scheduler in a DFS. Therefore in the formulation of MMGRCPSP problem (Figure 7), the objective function (11) maximizes the number of sorties flown during a period. Constraint type (12) enforces the renewable resources while constraint type (13) enforces the sortie production limit of each individual resource (nonrenewable resources). Expression type (14) enforces the lag time between predecessor and successor activities. Constraint type (15) is also different from Van Hove's MMGRCPSP problem formulation. It assures that each activity is processed once or it is not processed at all which means a mission is flown only once in a DFS. The binary decision variable, x_{imt}, is equal to 1 if activity *i* starts in period *t* and is executed in mode *m*.

The model generates an optimal schedule in terms of maximizing the sorties flown. However, it does not model replanning, student activity list and take off and landing times de-conflictions. Furthermore, it does not allow schedulers to make any interactions to the schedule before or after the schedule is generated. Also, the scheduler cannot insert any previously requested operational mission into the schedule. It has been used to solve a relatively small problem to optimality in less than 30 seconds with Excel Solver.

The number of available periods increases significantly if the duration of periods is decreased in a particular time window to get a better resolution in the model. Any increase in the number of available periods will increase the number of decision variables (number of decision variables = (number of activity) x (number of modes) x (number of periods)). This will increase the sizes of the resource matrices. As a result, the problem grows, becoming intractable for the standard Excel solver.

Parameters:

A the set of all activities (related to student *m*)

d the index of terminal activity

e_{im} the earliest completion time for activity *i* in mode *m*

ℓ_{im} the latest completion time for activity *i* in mode *m*

τ_{im} the duration of activity *i* in mode *m*

S_i the set of all generalized successors of activity *i*

Δ_{ijmn} the minimum lag between the start time of activity *i* in mode m and the start time of activity $j \in S_i$ in mode *n*

K the set of all double constrained resources

r_{imk} the amount of resource *k* required by activity when being executed in mode *m*

R_k the per period availability of resource *k*

N_k the total amount of resource k available

g the deadline of project under consideration

Variables:

x_{imt} 1 if activity starts in period t and is executed in mode m

$$\text{Maximize} \quad \sum_{m \in M_d} \sum_{i \in A} x_{imt} \qquad (11)$$

Subject to:

$$\sum_{i \in A} \sum_{m \in M_i} \sum_{t=\tau_{im}+1}^{j} r_{imk}\, x_{imt} \le R_k \quad \forall\, k \in K \qquad (12)$$

$$\sum_{i \in A} \sum_{m \in M_i} \sum_{t=e_{im}}^{j} r_{imk}\, x_{imt} \le N_k \quad \forall\, k \in K \qquad (13)$$

$$\sum_{n \in \mathrm{Mj}} \sum_{t=e_{jn}}^{l_{jn}} (\Delta_{ijm'n} - t)\, x_{jnt} + \sum_{t=e_{im'}}^{\ell_{im'}} t\, x_{im't} \le \sum_{m \in (M_i \setminus m')} \sum_{t=e_{im}}^{\ell_{im}} g\ x_{imt} \qquad (14)$$

$$\forall\, i \in A,\ \forall\, j \in S_i,\ \forall\, m \in M_i$$

$$\sum_{m \in \mathrm{M_i}} \sum_{t=e_{im}}^{\ell_{im}} x_{imt} \le 1 \quad \forall\, i \in A \qquad (15)$$

$$x_{imt} \in \{0,1\} \quad \forall\, (i,m,t) \qquad (16)$$

Figure 8. Complete MMGRCPSP for DFS

2.3 Scheduling in the 143rd Oncel Squadron Environment

The 143rd Oncel Squadron is a dual mode squadron training future F-16 pilots for TUAF. Both experienced instructor pilots and bandits staff Oncel. Bandits are the pilots chosen from the experienced pilots soon to become instructor pilots. Flight training at Oncel has a cyclical nature with a cycle time of approximately 6 months. SPs come biannually and each student class consists of 20 to 30 candidates.

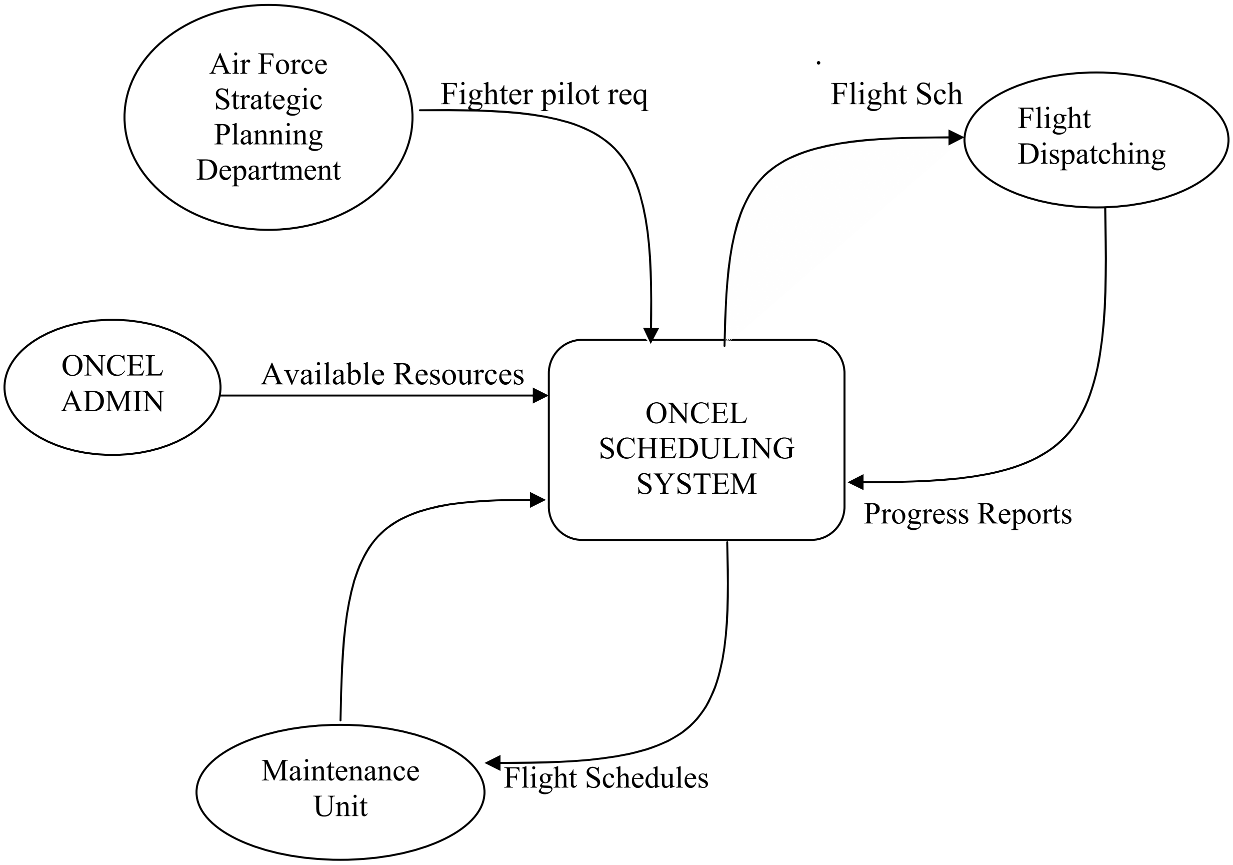


Figure 9. 143rd Oncel Squadron Scheduling System Context Diagram. [Evren,1999]

Although overlaps between consecutive student class training periods are allowed during the initial orientation and ground training phase, actual flight training does not overlap between classes.

A whole training period includes a sequence of precedence related events such as orientation, academics, avionics/cockpit familiarization training (AFT/CFT), pre-flight simulator sorties, and flight training. The number of aircraft assigned to the squadron is somewhere between 30-40. Some of these aircraft are two-seat (tandem) trainer models (F-16D) and are used extensively throughout flight training. The rest are single-seat (F-16C) models.[Evren, 1999].

Oncel scheduling system interacts with many external entities. When the data flow diagrams of Oncel are explored, it reveals some distinctive processes for the scheduling function. Two of these processes are long-term planning and short-term planning. Long-term planning is a time period of 6 to 8 months. Short-term planning comprises the proposed weekly schedules and daily flight schedules. However, daily flight scheduling is the center of attention for short-term planning in the scheduling department.

2.3.1 Scheduling Process

The objective of Oncel squadron's is to minimize makespan by maximizing the sorties flown in a day. A minimum makespan usually implies increased utilization of the resources. The experienced schedulers perform the scheduling processes manually. Everyday, schedulers face the demanding challenge of generating the draft schedule. An expert scheduler can quickly generate a draft schedule for a number of jobs in an hour. However, any change in the resource status can cause the scheduling processing to start over. After the draft schedule is generated on a chart, it has to be written on a spreadsheet and other entries such as aircraft allocation, call signs, mission frequencies, aircraft load and so forth has to be made by the scheduling NCO and maintenance people to draft schedule.

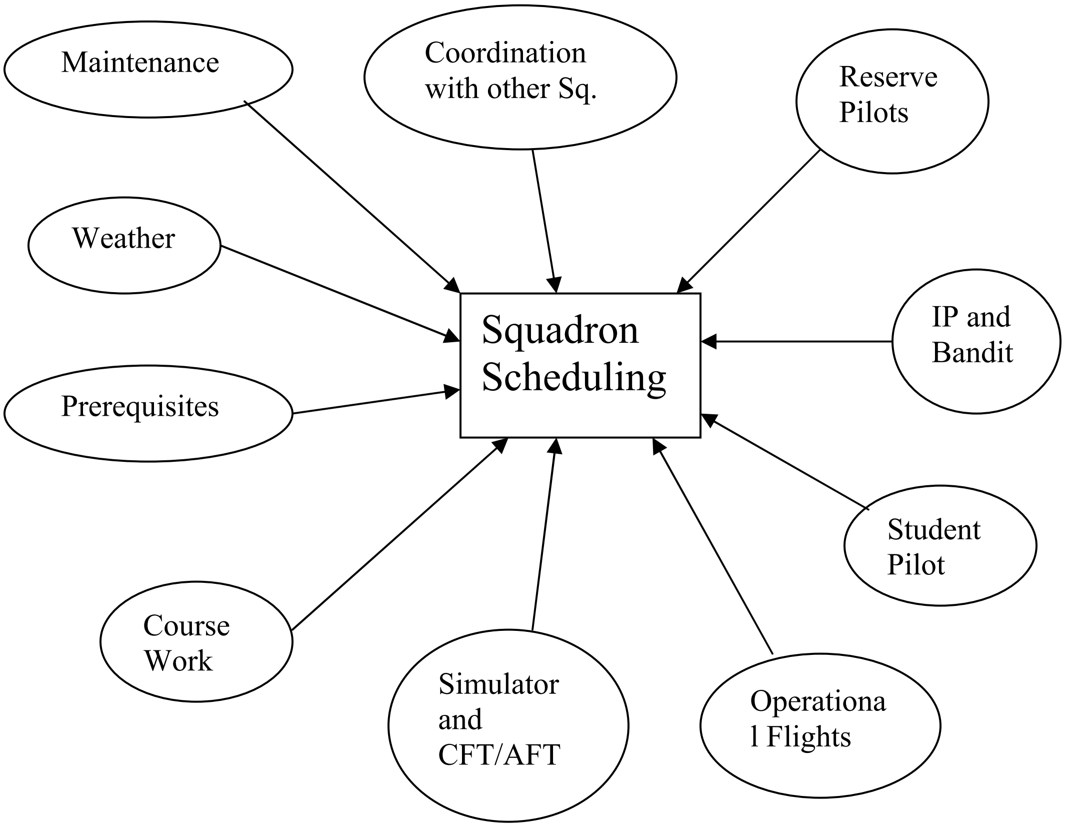


Figure 10. Squadron Scheduling Input Resources

"In practice, no schedule works out exactly as it was planned. This is true for variety of reasons, some of which can be anticipated" [American Institute of Certified Public Accounts, Inc, p.20, 1973]. Any change in the resource status after these entries have been made doubles the amount of time spent to generate the DFS. Therefore, any tool, at a minimum, needs to generate a whole schedule faster and is at least as efficient as the schedule produced manually.

2.4 Heuristics

"A heuristic is a technique which seeks good (i.e. near optimal) solutions at a reasonably computational cost without being able to guarantee either feasibility or optimality, or even in many cases how close to optimality a particular feasible solution is." [Reeves, p.6, 1995]

"Some problems have a combinatorial nature. This term is usually reserved for problems in which the decision variables are discrete- i.e. where the solution is a set, or sequence, of integers or other discrete objects." [Reeves, p.2, 1995].

Many combinatorial problems can be formulated in zero-one programming terms [Muller-Merbach, p3, 1981].

The design of heuristics requires decisions, and the decisions are choices among alternatives, which have to be explicitly available. Many of the problems for which no efficient converging algorithm exists are of a combinatorial in nature. In order to understand the functioning of heuristics, it is valuable to present their place within the system of the algorithms.

2.4.1 Heuristics in the System of Algorithms

Heuristics are a subset of algorithms. Therefore it is important to define the location of heuristics within the system of algorithms. Algorithms are the procedures used for solving a problem stated in mathematical terms.

> Most algorithms work iteratively, i.e. certain procedures are repeated several times. Iterative algorithms may not necessarily converge towards the sought solution. These are the algorithms, which will be called Heuristics. Even if the uncounted numbers of iterative algorithms differ from each other in many details, a general structure can be shown which

> represents the vast majority of the iterative algorithms, if not all. [Muller-Merbach, p.6-8, 1981].

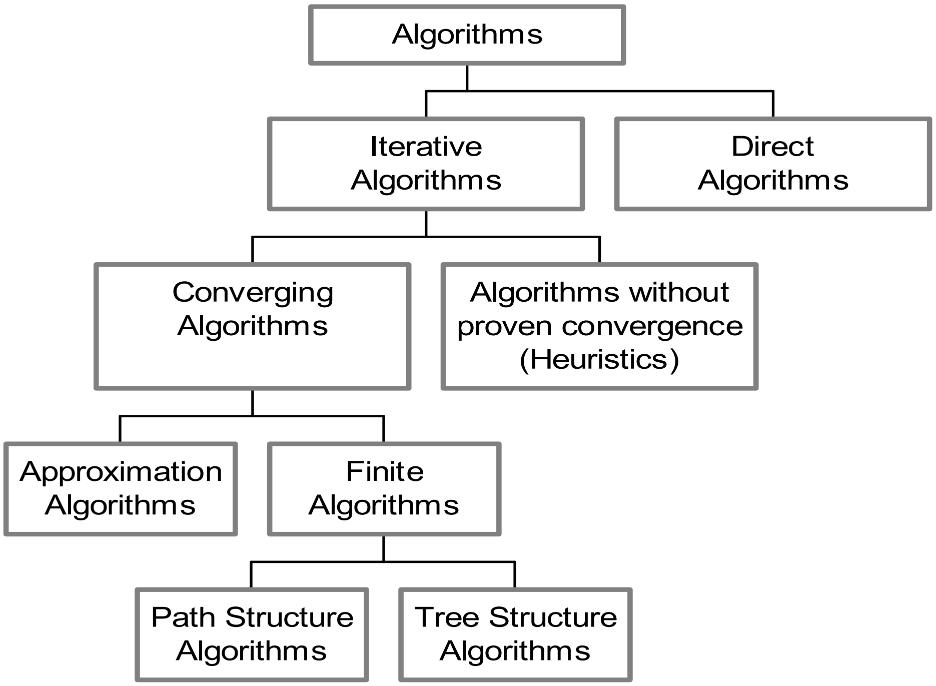


Figure 11. Tree with System of Algorithms [Muller-Merbach, p.6-8, 1981]

> A generalized neighborhood principle can describe this structure. Each iteration starts from a solution state. From here the candidates have to be determined and evaluated so that the solution state for a following iteration may be chosen. The set of candidates are called the neighborhood of a state. [Muller-Merbach, p.6-8, 1981].

There is not only one single neighborhood to a solution state. Instead, a hierarchy of neighborhoods could be identified for most iterative algorithms. Muller-Merbach states following four hierarchical levels which seem to be relevant for many of them.

- Neighborhood 1: Set off potential candidates

- Neighborhood 2: Set off considered candidates. For determining neighborhood 2, those potential candidates must be excluded in which they are obviously of no use
- Neighborhood 3: Set off accepted candidates. For determining neighborhood 3, all the considered candidates have to be evaluated, and those, which are not necessary, will be rejected. Only those, which seem to be necessary to find the solution will form the set of accepted candidates.
- Neighborhood 4: Set off selected candidates. This neighborhood 4 is only defined for heuristics. Out of the accepted candidates, some will be dropped due to the specific rules of the heuristics by which even the sought solution may be thrown away. The rest form the selected candidates.

Each iterative algorithm consists of a sequence of iterations. Each iteration starts by choosing a solution state. From there, the neighborhoods (from level 1 to level 3 or 4, respectively) are determined. Then, the next iteration begins with choosing a solution state. The whole procedure stops if either there is not any solution state remaining whose neighborhoods were not yet determined or the neighborhood is empty for the last and only solution state under consideration [Muller-Merbach, p.6-8, 1981].

> In heuristics, it is common that all but one of the selected candidates of each neighborhood is dropped since the most inefficient algorithms have a tree structure. This leads to a path structure, which would be the easiest organization of heuristics but not necessarily the most effective one. Therefore sometimes it can be advantageous to follow parallel path as well. [Muller-Merbach, p.6-8, 1981].

Often there is not just one heuristic, which is applied to a certain problem, but several. In this case, a procedure is required that determines how the individual heuristics succeed one another.

2.4.2 Types of Heuristics Methods

Heuristics are often simpler to understand and comprehend than most of the mathematical models. They give an insight to the problem. Silver, Vidal and Werra defines six categories of heuristic methods [Silver, *et al*, p. 153-162, 1980]. They also state that the categories are not meant to be mutually exclusive and it often makes sense to blend two or more types in the solution of a particular type of problem.

- Decomposition methods: Larger problems are broken down into smaller pieces. After each small problem is solved separately, the solutions are combined to obtain the overall solution for the larger problems.
- Inductive methods: the solution properties and heuristics characteristics obtained from smaller instances are generalized
- Reduction methods: the size of the problem is reduced so that the algorithms work more efficiently
- Model manipulating methods: prior to solution, the nature of the mathematical model is changed.
- Constructive methods: used to build a feasible solution. Generally these are single solutions, deterministic in nature, and the greedy type of heuristics. It has two types of approaches to build a feasible solution, primal approach and dual approach.

- Local improvement methods: used to move in some improving direction at some feasible (or infeasible) solution [Silver, *et al*, p. 153-162, 1980].

> A good heuristic should be simple to understand. It should have a reasonable storage requirement. It should be fast and it should have an accurate solution. A good heuristic should give good answers most of the time and there should be low variance about these answers. It should be able to handle a wide variety of problem instances reasonably well with little to no performance differences due to minor input changes. [Class Notes, Oper-623, 2001 fall].

2.5 Object Oriented Programming

Programming languages must be considered to implement any heuristics, or pre defined dispatching rules. To select the right programming language, considerations of the selection must be based on the criteria of their availability as well as being easy to learn and use. The majority of the desktops computers in the scheduling division use a version of the Microsoft Windows operating system. "Since Microsoft also develops the MS Office Suite on the foundation of Visual Basic engine, they can build enhancements and attachment modules into the application to solve specific problems, and is assured a very high probability of error free integration" [Nguyen , p.21, 2002].

MS office products such as Access, Word, and Excel have become the main word processor, database and spreadsheet in the majority of offices and homes. The required software is already present in the office documents because these come already pre installed with the computers when they are first purchased. Oncel squadron scheduling division used to generate and publish the schedules with either a paper chart or with MS Excel spreadsheet designed similar to the chart.

Visual Basic is used for these compelling reasons over Java and other object oriented programming languages.

In VBA, the attributes of an object are called *properties*: the size property, the color property. In addition, each property has a *value* for any particular car. For example, the car might be white and it might have four doors. In contrast, the things can be done to an object are called *methods*: the drive method, the park method. Methods can take qualifiers, called as *arguments*, which indicates how a method is carried out [Albright, p.7, 2001].

Some of the most common objects in Excel are ranges, worksheets, charts and workbooks. For example, consider the single-cell range B5. This range is considered a Range object. It has a Value property: the value (either text or numeric) in the cell. A Range object also has methods. For example, a range can be copied. Copy method takes the destination as its argument.

There is an object hierarchy in Microsoft Excel Objects. At the top of the hierarchy is the Application object. This refers to Excel itself. One step down from application is the Workbooks collection. One step down from Workbook is the Worksheet objects and the other objects follow it. [Albright, p.8-9, 2001]

2.6 Summary

This chapter covered the topics related to scheduling theory. In addition to a review of the pertinent literature on flight scheduling, the chapter also provided the background on the scheduling environment of 143rd Oncel Squadron with the pertinent literature and the 143rd Oncel squadron background established, Chapter 3 presents the methodology for solving the squadron scheduling problem.

CHAPTER 3. METHODOLOGY

Overview

This chapter describes how the topics and methods from Chapter 2 were applied to the flight-scheduling problem. This chapter is partitioned into five distinct areas: scheduling goals and objectives, the scheduling model and problem characteristics, construction heuristics for initial solutions, the new shifting bottleneck heuristic, and the new order heuristic.

3.1 Scheduling Goals and Objectives

The squadron schedulers at Oncel produce the DFS to meet certain goals and objectives. The schedules are utilized to ensure students receive the necessary instruction to meet training goals and to graduate on time.

In addition to the SP training, the scheduler ensures that IPs and Bandits are assigned to missions and additional duties, such as Runway Supervisory Unit (RSU) or Range Unit, while meeting squadron policy requirements. This is not a hard constraint for the scheduling problem on a daily basis, but if not implemented in to the schedule, it may cause difficulties over a longer period. For this reason it has to be accounted for early on.

If an IP is assigned to two different student activities in the same flight day, then he has to prepare and give two different briefings and debriefings in addition to flying two different missions. This significantly increases the workload of an IP in a flight day. The instructor may very quickly fatigue if this is done several times in the same week.

Eventually, it may result in future sortie losses. Therefore, a third goal is to minimize the workload of IPs by assigning them to the same kind of activities, i.e. missions, in the DFS as much as possible.

Furthermore, the scheduler must account for, on an individual basis, each activity performed by any pilot during the day. The objective of this goal is to balance the workload among the assigned squadron pilots based on squadron policy.

Students may have the opportunity to improve their flying skills by flying more sorties than the minimum number required by the syllabus given favorable weather and efficient scheduling during the B course. These extra flights should be distributed equally among the SPs to provide additional training while maintaining balance.

The overall objective for the squadron scheduler is to establish a robust schedule that will satisfy all of these varied training goals. The objective of this thesis is to provide a scheduling tool that reduces time required to build a robust DFS.

3.2 Scheduling Model and Problem Characteristics

Before looking into the model, a high-level review should be made to the scheduling process. The scheduler must ensure the availability and the amount of resources before producing a draft schedule. As mentioned in Chapter 2, these resources are categorized into two groups, renewable and nonrenewable resources. The scheduler should know the amount of the renewable and nonrenewable resources available before making the next day's draft schedule. A mission or missions may have to be rescheduled or cancelled depending on the availablity of the nonrenewable resource if the schedule is produced only relying on the amount of renewable resources.

The scheduling process in Oncel can be summarized by the following 6 steps.

1. Determine the potential flyable missions for the next day.
2. Receive data: IP, SP and Support Pilot availablity, maintenance aircraft availability, sortie requests for the operational flights.
3. Assign IPs or Bandits for the duties.
4. Generate a draft schedule.
5. Confirm it with other squadrons, IPs, simulator and maintenance
6. Prints the schedule

To produce a robust schedule, the scheduling environment has to be understood in terms of its dynamic changes, scheduling requirements and other scheduling related constraints. Operational, maintenance, and weather cancellations may occur at any time. Requirements or duty changes between the squadrons happen frequently. In addition, the squadron schedulers rotate periodically among themselves. These changes make for a dynamic training environment at Oncel that often requires scheduling and rescheduling.

Each type of cancellation affects the schedule in different ways. Weather plays an important role in training squadron schedules. The weather may restrict some or all missions due to cloudy and/or low visibility conditions. Scheduling and rescheduling must respect these weather conditions.

Operational cancellations are related to the squadron. A SP or IP might become ill, and, if there is no suitable substitute for them, the mission is cancelled. If a mission is cancelled due to the aircraft performance and/or availability then it is called a maintenance cancellation. Similarly, if alternate aircraft are not available, the mission is cancelled. Of these cancellations, those related to weather have the most affect on the schedule. All or most of the missions must either be cancelled or changed when a cancellation due to weather occurs. These dynamic changes often require small readjustments to the original schedule to keep the goals and objectives satisfied. Sometimes the multiple goals and objectives of the DFS conflict with each other. When conflicts occur in the DFS, the aircraft sortie schedule receives the highest priority when the schedules are in conflict.

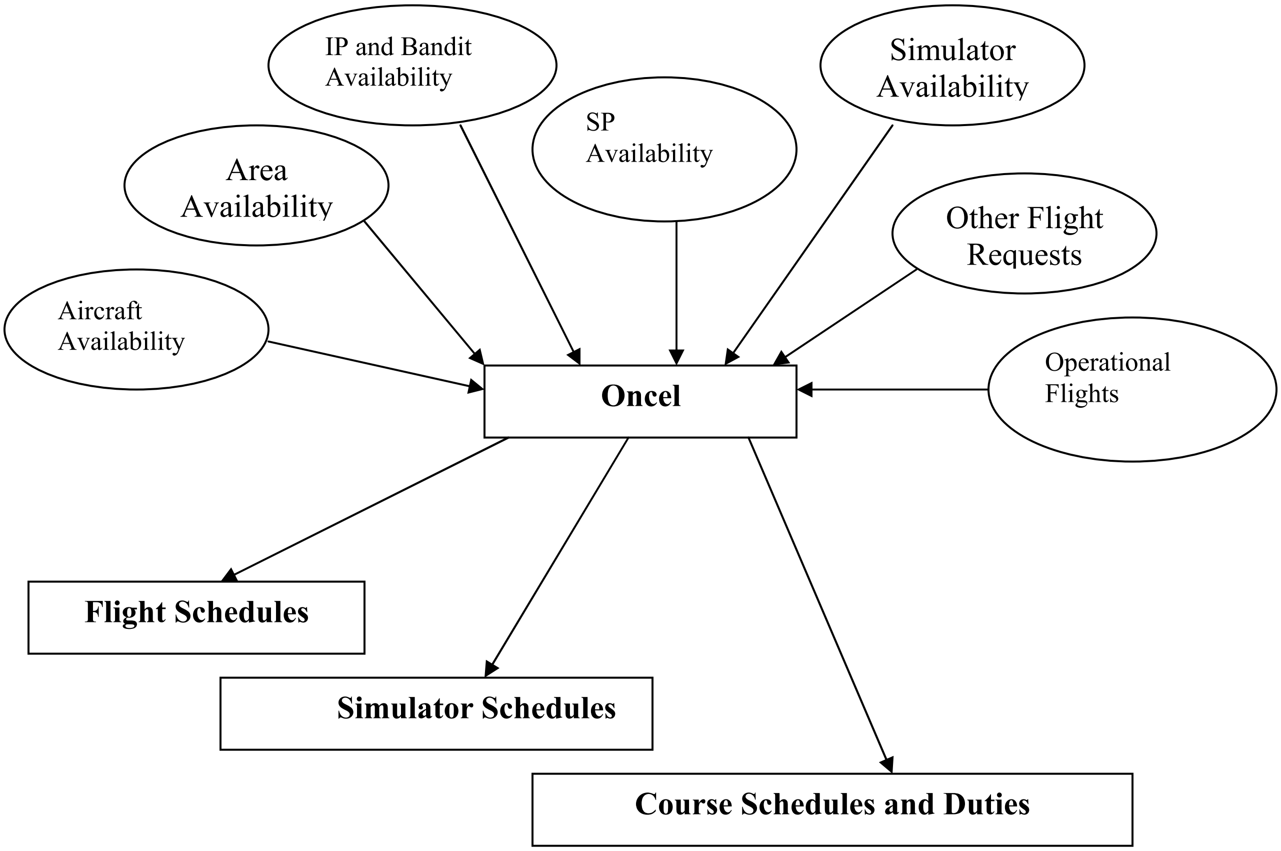


Figure 12. Squadron Scheduling Inputs and Outputs

The squadron scheduling process produces the flight, simulator and course schedules (Figure 12). These activities have different priorities and different resource requirements. Because there is only one simulator available for Akinci AFB, a simulator mission has priority over a course unless the course is a prerequisite for the simulator mission. The courses may be scheduled any time during the day when every student is available, but the students cannot all fly the simulator at the same time. Missions have precedence over the simulator and courses. Every student should be available when a course is scheduled so as not to duplicate effort. Generally the flight missions require good weather whereas simulator missions can be flown and courses can be taught in any weather. For this reason, especially in wintertime, a flight mission has priority over a simulator mission unless the simulator mission is a prerequisite for the scheduled flight mission.

3.3 Construction Heuristic For The Initial Solution

As this research concentrates on producing a robust DFS, it is important to explore some heuristics to generate better initial solutions. This initial solution should include precedence constraints and maintain feasibility. Recall the Largest Number of Successors (LNS) rule from Section 2.1.4. Under the LNS rule the job with the largest number of successors in the precedence constraint graph has the highest priority. For example, suppose student A is flying mission 5, student B is flying mission 10 and student C is flying mission 12 out of 62 flight missions. Given the respective flight status, student A, with 57 missions remaining, obviously has a higher priority than the other students under the LNS rule. Student B has a higher priority than student C. Another implementation of LNS rule occurs when two activities to be completed by student A are intended to be scheduled in the same day, such as a simulator mission followed by a flight mission or a flight mission followed by a simulator mission. In this case the first activity of student A has a higher priority than the other activities. An optional implementation involves a flight mission followed by a simulator mission. This implementation is only allowed when there is a problem with maintaining the course completion timeline or for other reasons related to time and bottleneck. The former has the priority over the latter if both are implemented on the same day.

Recall the LFJ and LFM heuristics from Section 2.1.4. If an activity is scheduled for a student, then the mode to execute the mission is chosen according to the LFJ-LFM heuristic. Should a tie occur under the LNS rule between two activities then LFJ rule will select the activity, which has less execution modes than the others. This is analogous to selecting job *i* which can be processed in a fewer number machines than job *j* if there are two jobs, job *i* and job *j*, ready to be processed on a subset of parallel machines. The LFM rule chooses the mode of the scheduled activity depending on the number of assets available in the squadron. Consider the earlier example in the previous paragraph where there is student D in the mission order list (MOL) and he is also ready for mission 5. According to the LNS rule student A and student D have the same level priority because they both have 57 missions remaining as the successors of this 5th mission. The tie is then broken according to the number of modes available for these missions. If the 5th mission of student A can be processed in two modes and the 5th mission of student D can be processed in 3 modes, then student A is scheduled first in the DFS. If the numbers of modes are equal for student A and student D, the tie is broken arbitrarily.

After scheduling student A as the first SP to fly mission 5, a decision has to be made about how to process this activity. At this point the LFM rule decides which mode is to be selected. Each mode represents a vector of resources, which are needed to accomplish a specific activity. Hence, LFM selects the mode that uses the most available asset on the base. In this instance if mission 5 of student A has two modes, mode 1 and mode 2, then each mode's resource use is compared to each other. If the squadron has 10 F 16C and 5 F 16D aircraft and mode 1 needs 2 F 16C and mode 2 needs 1 F 16C and 1 F 16D then mode 1 is selected by the LFM rule.

The DFS is closely related to a machine shop with the following features: unrelated machines in parallel, precedence relationships, each job can be processed on a subset of available machines like flexible job shops and preemption is not allowed at all. The objective is to minimize the makespan (the amount of time it takes to process all the activities) for a group of students. Maximizing the daily produced sorties provides the minimum makespan if the problem is handled with a myopic approach. This suggests a heuristic to generate an initial solution for assigning assets to activities. The heuristic is combination of the LNS rule, the LFJ rule and the LFM rule (referred as LNS-LFJ-LFM rule).

There is another important consideration before the steps of the heuristic can be outlined, that being lag times. The LNS-LFJ-LFM rule provides an initial feasible solution when it is implemented with the following lag time constraints. In the previous example four students are to be scheduled and the order is given as student A first, followed by student D, student B and then student C. In this example student C cannot take off until the rest of the students take off and student B cannot take off until student A and student D take off. For this research, missions must be flown in the order of LNS-LFJ-LFM rule selection. This gives rise to a minimum lag time, which may constrain the take off time for the successor missions. This concept is illustrated with a simple example (Table 2).

Table 2. Example Mission Data

Order of Missions (in pairs)	Take off Times	Landing Times
Student A-mission 5 (A5)	10:00	11:15
Student D-mission 5 (D5)	10:10	11:25
Student B-mission 10 (B10)	10:20	11:25
Student C-mission 12 (C12)	10:40	12:00

Table 2 contains the mission data from the previous example. Landing times are given according to the mission durations of each mission. If the time window starts at 10:00 the first mission in the order list takes that slot. To schedule the consecutive missions in the order list determined by the LNS-LFJ-LFM rule, a lag is necessary between the take off times. This lag is dependent on the mission's characteristics more than the squadron policy. Let S_i be the start for mission *i*. The constraint may be formulated as follows for each successive mission in DFS:

$$S_{D5} \geq S_{A5} + 10$$

The 10 in the inequality is a notional lag value associated with the general precedence constraints of take off times. Suppose mission A5 takes off at 1000. The constraint on the successor is found as follows:

$$S_{D5} \geq S_{A5} + 10 \rightarrow \qquad S_{D5} \geq 1000 + 10 \rightarrow \qquad S_{D5} \geq 1010$$

The lag times between the take off and landing times of each mission are also implemented in the model. A mission cannot take off at the same time when another mission lands. Let S_i be the start for mission *i* and C_i be the finish time for mission *i*. The constrain may be formulated as follows for each mission pair in DFS:

$S_{D5} \geq C_{A5} + 5$ or $S_{D5} \leq C_{A5} - 5$

Negative 5 and 5 are the notional lag values associated with the general precedence constraints of takeoff and landing times. Consider the previous example and assume mission A5 lands at 1115. The constraints on the successors are found as follows:

$S_{D5} \geq C_{A5} + 5 \rightarrow \quad S_{D5} \geq 1115 + 5 \rightarrow \quad S_{D5} \geq 1120$ or

$S_{D5} \leq C_{A5} - 5 \rightarrow \quad S_{D5} \leq 1115 - 5 \rightarrow \quad S_{D5} \leq 1110$

The change is not necessarily applied to only the take off time. In most cases, if there is a conflict between the takeoff time of one mission and the landing time of another mission, then generally the landing time is altered a few minutes to generate a feasible solution.

Since two missions cannot land at the same time, there has to be a lag time between landing times of each mission pair in the DFS. Let C_i be the finish time for mission *i*. The constraint may be formulated as follows for each mission pair in DFS:

$C_{D5} \geq C_{A5} + 3$ or $C_{D5} \leq C_{A5} - 3$

Negative 3 and 3 are the notional lag values associated with the general precedence constraint of landing times. Consider the previous example and suppose mission A5 lands at 1115. The constraints on the successors are found as follows:

$C_{D5} \geq C_{A5} + 3 \rightarrow \quad C_{D5} \geq 1115 + 3 \rightarrow \quad C_{D5} \geq 1118$ or

$C_{D5} \leq C_{A5} - 3 \rightarrow \quad C_{D5} \leq 1115 - 3 \rightarrow \quad C_{D5} \leq 1112$

In summary, lag time is comprised of three types of constraints related to take off and landing times that make the initial solution feasible. These are the lag time between the take off times, between the take off of one mission and landing of another mission and between the landing times of the missions. In Table 2, student D and the student B are landing at the same time, therefore this schedule is infeasible. The following schedule was developed with a predetermined minimum lag times between landings of two missions (Table 3)

Table 3. Example Mission Data with Lag Enforcement

Order of Missions (in pairs)	Take off Times	Landing Times
Student A-mission 5 (A5)	10:00	11:15
Student D-mission 5 (D5)	10:10	11:25
Student B-mission 10 (B10)	10:25	11:30
Student C-mission 12 (C12)	10:40	12:00

Notice that the take off time of the mission B10 is altered for 5 minutes to provide a difference between landing times of missions B10 and D5. There is another lag time constraint, which is related to resources that has to be implemented in the model. This fourth lag time is associated with the areas and it ensures that a successor mission does not ingress into an area before the predecessor egresses from that same area.

To provide the lag between two consecutive missions into the same area, two generalized precedence constraints must be defined, one for the predecessor pair and one for the successor pair. Let S_i be the start for mission *i*. For a particular mission, 30 is the notional lag values associated with the general precedence constraint of area times. The constraints may be formulated as follows:

$S_{D5} \geq S_{A5} + 30$

$S_{D5} \leq S_{B10} + 15 + 30$

Observe that generally the start times of the successors are constrained by the start time of the predecessors but in some cases the start times of the successors are forced by the finish time or area egress time of the predecessor activities. Solutions developed in this research implement lag times into the model and suggest take off times for each activity.

Additionally, the fighter training squadron scheduling support tool (FTSSST) can assign persons for duties such as supervisory of flight (SOF) and runway supervisory unit (RSU). If there is an AG mission in any period of the DFS, the FTSSST must assign a person to the range during that specific period. The FTSSST keeps track of every duty performed by the squadron on an individual basis and records this information on a separate spreadsheet. The assignment of additional duties is done according to the squadron policy, which is comprised of military ranking and a pilot's current attributes, e.g. BP, IP, sick or not sick. The scheduler can interact with the FTSSST anytime during the duty assignment phase and change the names and positions of the assigned pilots before or after the FTSSST.

The scheduler, if needed, may also change the suggested takeoff and landing times. The scheduler can change the order of the missions in the MOL or in the DFS as well as change the areas used for the missions.

If an activity must be flown at a certain time, the software lets the scheduler insert the mission in the DFS. Additionally, he can insert this mission into the MOL and the mission is scheduled during the time requested. The FTSSST model also allows the scheduler to specify the resource use for a specific mission. If any change occurs in the resource use of a mission due to a syllabus change, the FTSSST is flexible enough to schedule the mission after the predefined resource vector change has been defined. Recall the previous example from Section 3.3. If the mission 5 is an F 16C and F 16D mission then the resource vector is [1 1] for aircraft use. If the mission 5 is changed as to be flown by two F 16D aircraft then the resource vector will be [0 2]. After this change is made to the resource vector, the FTSSST will generate the DFS according to the new syllabus.

Some of the missions in the syllabus require four aircraft. To reduce IP and BP workload, these missions are flown simultaneously by 2 SPs. For this reason the SPs must fly in the same formation so that the squadron resources are most effectively utilized. The missions are not scheduled until they can be placed successfully within the MOL or user interaction forces them into the existing schedule.

During some phases of the training syllabus, a condition known as a bottleneck might occur. That is, all of the successor missions will be on hold until the predecessor missions causing the bottleneck are flown. Hence the Smallest Number of Successors (SNS) rule (under this rule, the job with the fewest number of successors in the precedence graph has the highest priority), the LNS rule and the General Mission (GM) order heuristic, described in the next section, are used as MOL and the flight scheduling shifting bottleneck heuristic (FSSBH) is applied to decrease the makespan and increase the number of sorties flown over all.

3.3.1 General Mission (GM) Order Heuristic

The GM order heuristic is problem specific. The GM rule selects the LNS mission order or the SNS mission order according to the FSSBH. Given the selected MOL, the remaining missions of the SPs are compared. Based on this comparison and the current phase of the flight program, the GM can re–prioritize some SPs by moving them and their missions up or down in the selected list or does nothing at all. The AG phase provides an example of when this heuristic is applied. During the AG phase, two consecutive missions going to AG range must have a minimum lag time of 30 minutes. To be able to schedule other missions within this 30-minute period, the GM rule inserts two other types of missions in between the range missions if they are available on the mission list. The GM rule also arranges the MOL according to the missions, which have to be flown as a four-ship flight for AG missions. The scheduler can still interact with the order and at anytime he can change the order of missions, insert a new mission to the list or cancel a mission from the list.

3.3.2 The Construction Heuristic

The heuristic presented in Table 4 orders each mission in association with one of these predefined rules and scheduler’s choice determines the type of initial solution. This construction heuristic generates an initial schedule according to the scheduler’s choice. The scheduler can choose any of these three flight order rules to generate the DFS. If the scheduler chooses the FSSBH, then the dispatching rule is going to be selected automatically. The FSSBH will generate the entire DFS unless directed otherwise by the decision makers.

Table 4. Feasible Initial Solution Construction Heuristic

1. Persons are assigned to duties
2. Scheduler selects the MOL
3. The missions are ordered according to the selected priority rule
4. If needed, "must-fly" student missions are prioritized in the MOL
5. Ties are broken according to the LFJ-LFM first rule
6. DFS is generated and if needed missions can be altered or a new mission can be inserted
7. Other resources for the mission are recorded on the DFS
8. Simulator schedule is produced (if required)
9. Course work is scheduled.

The squadron scheduler can also make some iterative adjustments to the initial schedule by changing the status of the resources until a better schedule is found. If all of the Air-to-Air (AA) configured F-16D model aircraft are used in the DFS, then the scheduler can change the configuration of one F-16D model aircraft from AG to AA and rerun the FTSSST to see if any improvement is seen in the number of sorties produced in the DFS. He can also make these kinds of changes to some other resources to see their effect on the schedule.

Configuration determination also reduces the maintenance time and aircraft parts wear and tear.

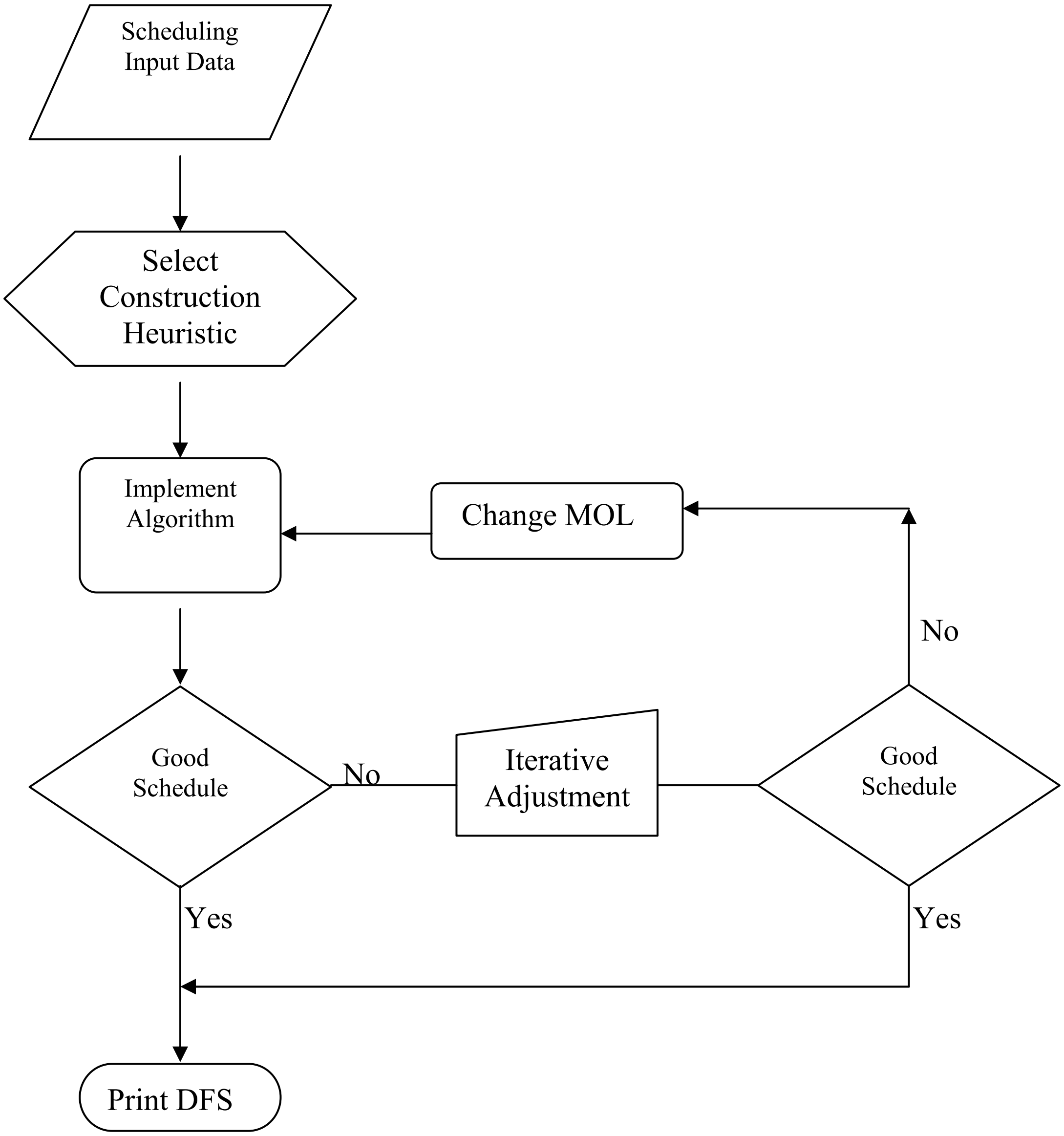


Figure 13. Software Implementation of Construction heuristic

In Figure 13 the overall flow of generating an acceptable DFS is given. Input data is obtained from a database produced in Microsoft Access. The scheduler then updates this data for any changes. After every resource status is entered to the tables, the scheduler can chose the dispatching rule to produce the DFS.

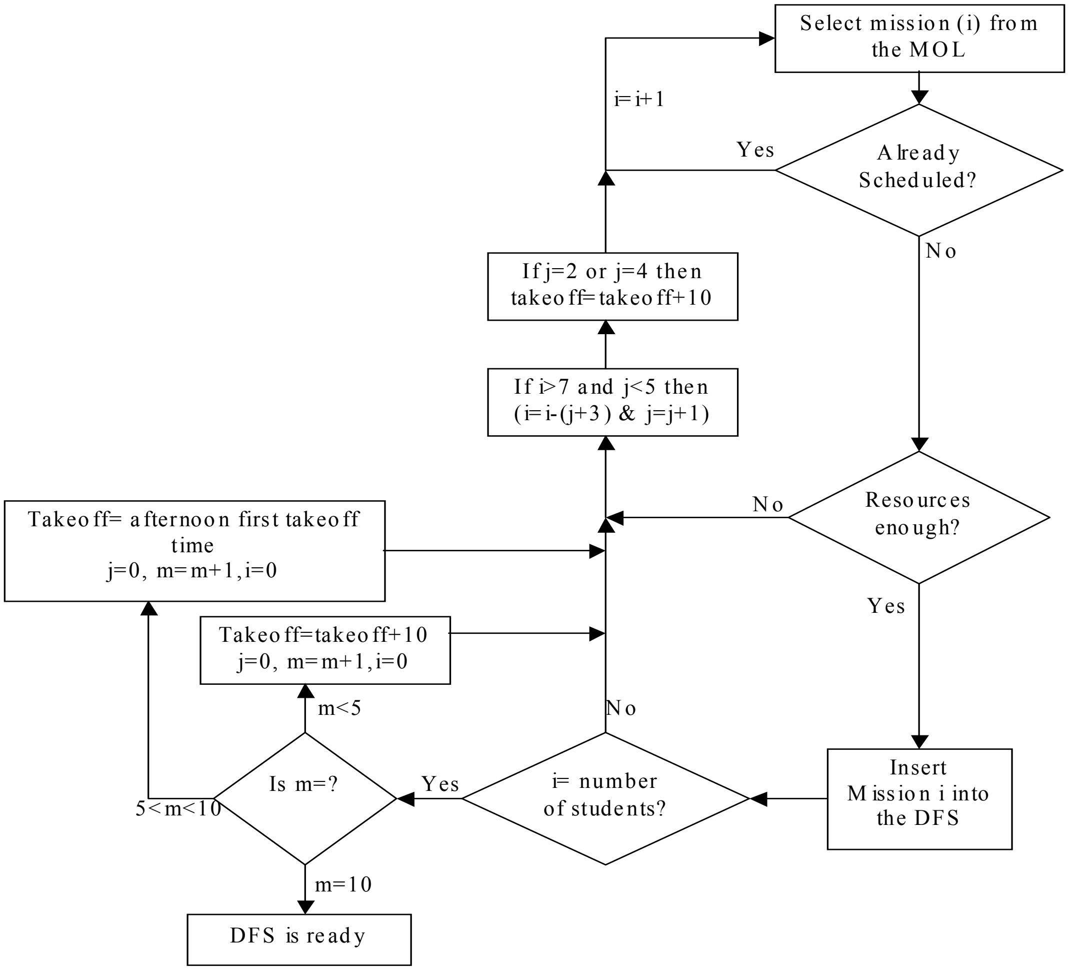

Select mission (i) from the MOL
i=i+1
Yes
Already Scheduled?
No
If j=2 or j=4 then takeoff=takeoff+10
If i>7 and j<5 then (i=i-(j+3) & j=j+1)
No
Resources enough?
Takeoff= afternoon first takeoff time j=0, m=m+1,i=0
Yes
Takeoff=takeoff+10 j=0, m=m+1,i=0
m<5
No
Is m=?
Yes
i= number of students?
Insert Mission i into the DFS
5<m<10
m=10
DFS is ready

Figure 14.Draft Schedule Heuristic Flowchart for a day

Based on his experience, the scheduler may be able to determine a schedule that is more effective in the current operational setting. If so, the scheduler may either change the order of the missions in the selected MOL or select a different MOL and redo the steps until an acceptable DFS is produced.

If the DFS is producing enough SP missions as preplanned but there are still non-binding resources then the scheduler can make iterative adjustments, as previously mentioned, to increase the number of sorties produced. When a “good” DFS is generated, it is printed as the last step.

To build the draft schedule, a construction algorithm is also implemented (Figure 14). In the 1st step FTSSST selects the first unscheduled mission from the MOL. Then, the selected mission is scheduled according to the resource availability. If it is not scheduled due to insufficient resources, the FTSSST picks up the next unscheduled mission in the list. Generally number of feasible sorties assigned in a “GO” is 8, hence, after the 7th mission, the algorithm allows the FTSSST to schedule previously skipped missions due to inadequate resources. After the second and fourth pass the takeoff time is increased by 10 minutes. This allows the FTSSST to schedule two consecutive AG missions in the DFS. The FTSSST adds an extra lag with a notional value of 10 minutes to takeoff time after the 6th and 12th scheduled missions. This extra time inserted in the schedule provides a chance to fly the missions for the fallback flights (i.e., delayed) due to maintenance problems. The FTSSST goes back to the first mission in the MOL and recheck every unscheduled mission to see whether it can be scheduled or not for five times. After the fifth pass, the second go is scheduled with the same algorithm.

3.3.3 FSSBH for FTSSST

As previously noted, the B course is a 62-sortie flight-training program that consists of three different flying phases. The first phase of the B course is comprised of two sub phases, AA phase is the second phase of the B course consisting of three sub phases and the third phase of the B course is AG comprising six sub phases. IQ and TQ check rides are the two sorties that have to be flown by an SP with success in order to graduate from the B course. During the planning horizon, depending on the current phases of the flight program, some resources will be scarce while other resources will be plentiful.

To be able to use these resources efficiently, Oncel's syllabus must be studied closely. In the first phase of the flight program, the F-16 D model is a scare resource, and, at times, the areas (i.e., range). As the phases of the flight program advance, the F-16D model is not the only scarce resource. The AG range becomes the most limiting resource for the number of sorties produces in the DFS during the third phase. For example, in the AG phase, if all of the missions in the MOL require the range, then the missions in the DFS must be scheduled with 30 minutes takeoff intervals, and the 10th and 20th minute takeoff times are lost because of the flyable mission unavailability in the MOL. This same type of bottleneck occurs if all the SPs fly basic training (BTR) missions. In this case, the squadron will use all of the available F-16D models but none of the available F-16C models. Therefore the FSSBH is defined to shift the bottlenecks caused by these scarce resources to increase the number of missions produced in DFS. The implementation FSSBH depends on the phase of the flight and the number of the SPs in the B course.

3.4 Summary

This chapter presented the methodology for solving the squadron-scheduling problem. The methodology allows the scheduler to interact with the FTSSST in any phase of the scheduling algorithm. The methodology lists the remaining missions in three different order rules and inserts these missions into the DFS using the construction heuristic described in Figure 14.

In summary, before the DFS is generated, the scheduler:

1. Checks the Availability of
    - IPs
    - SPs
    - Bandits
    - Aircraft
2. Enters the times into the data entry table for
    - The first go
    - The second go
    - The third go, that is, the night flight
    - The Air to Ground range
    - RSU duty
    - SOF duty
    - Area availability
3. Assigns people to
    - RSU
    - Range
    - SOF
4. Selects a MOL

5. Generates the draft schedule or full DFS
    - If the draft schedule is produced, then completes the rest of the schedule
6. Selects the SPs for simulator schedule and makes the simulator schedule

And after the DFS has been generated and the flights are realized, the scheduler:

1. Records the IP activities
2. Records the SP activities
3. Records the DFS
4. Clears the DFS

At the end of the whole cycle, the scheduler re-starts from the beginning for the next day's DFS.

Chapter 4 details how the methodology was tested and the results of this testing. The chapter also contains a case study problem that resembles an ongoing case in Oncel. The case study is used to produce a DFS for each day within a flight program and to demonstrate the implementation of the FSSBH and to compare the utility of the FSSBH to existing flight schedules.

CHAPTER 4. ANALYSIS AND RESULTS

General

This chapter covers the analysis of the schedules generated for a simulated set of environment conditions. The first section sets up a notional B-course training program according to the syllabus currently used by the 143rd Training Squadron. The notional training program uses the same number resources (aircraft, IPs, Bandits and SPs) as the Oncel scheduling division has available. Physical outlook and the performance of the software are analyzed in the second section. The third section analyzes and summarizes the notional training program under the LNS and SNS first rules as compared to the current syllabus. This section also analyzes and summarizes the notional training program under the FSSBH. Hence the effect of employing a shifting bottleneck heuristic will be demonstrated. This chapter also presents the outcome of applying a relaxed integer linear programming (ILP) formulation to several MMGRCPSP sets for determining the performance of the construction heuristic.

4.1 Notional Schedule Setup

A notional training program is created to test the FTSSST for analysis. The notional training program, ideally, should represent reality; therefore, as many of the features and characteristics of the real system as possible are included. One of these features involves the amount of resources available to a scheduler in Oncel. For this reason the notional schedule has 22 SPs, 24 IPs and 6 Bandits. Support pilots from headquarters and from other resources are not included in the notional training program. Additionally, the same number of F-16C and F-16D aircraft that Oncel has available is utilized by the notional training program. The notional training program also includes the same number of areas available for each day for the DFS as Oncel has available for real scheduling. Before the SPs start the B course, the scheduling division looks out 6 months to forecast the graduation time of the beginning class according to the available resources at hand, and by predicting their future availability biannually. For example, most of the IPs and Bandits take leave for 15 to 20 days during

Table 5. Resource Availability

DATES			Planning Factors			
			IPs	Bandits	F-16D	F-16C
19 Aug	-	23-Aug	18	5	8	26
26 Aug	-	30 Aug	18	5	8	26
02 Sep	-	06 Sep	23	5	8	26
09 Sep	-	13 Sep	23	5	8	26
16 Sep	-	20 Sep	22	6	8	26
23 Sep	-	27 Sep	22	6	8	26
30 Sep	-	04 Oct	22	6	8	26
07 Oct	-	11 Oct	22	6	8	26
14 Oct	-	18 Oct	22	6	8	26
21 Oct	-	25 Oct	22	6	8	26
28 Oct	-	01 Nov	22	6	8	26
04 Nov	-	08 Nov	22	6	8	26
11 Nov	-	15 Nov	20	2	8	26
18 Nov	-	22 Nov	20	2	8	26
25 Nov	-	29 Nov	20	2	8	26
02 Dec	-	06 Dec	26	2	8	26
09 Dec	-	13 Dec	26	2	8	26
16 Dec	-	20 Dec	26	2	8	26
23 Dec	-	27 Dec	26	2	8	26
30 Dec	-	03 Jan	26	2	8	26
06 Jan	-	10 Jan	26	2	8	26
13 Jan	-	17 Jan	26	2	8	26
20 Jan	-	24 Jan	26	2	8	26
27 Jan	-	31 Jan	26	2	8	26
03 Feb	-	07 Feb	26	2	8	26

summer time, and there are exercises Oncel has to support. The scheduler has to implement these absentees into the resource table so that the forecast is realistic. A resource availability table was developeed for the notional schedule with the same numbers the Oncel squadron has for the next 6 months (Table 5).

Table 6. Weekly Planning Student Sortie Requirements

B Course Flying Weeks	WEEKLY PLANNING (DAILY SORTIES)						
	MONDAY	TUESDAY	WEDN	THURSDAY	FRIDAY	SORTIES BY WEEK	Cumulative TOTAL SORTIES
1	10	10	10	10	10	50	50
2	10	10	10	10	0	40	90
3	10	10	10	10	10	50	140
4	10	10	10	10	10	50	190
5	8	10	12	12	12	54	244
6	12	12	12	12	12	60	304
7	12	12	12	12	12	60	364
8	12	12	12	12	12	60	424
9	12	12	12	12	12	60	484
10	12	12	12	12	12	60	544
11	0	0	12	12	12	36	580
12	12	12	12	12	12	60	640
13	12	12	12	12	12	60	700
14	12	12	12	12	12	60	760
15	12	12	12	12	12	60	820
16	0	0	0	0	0	0	820
17	12	12	12	12	14	62	882
18	14	14	16	16	14	74	956
19	8	8	10	8	8	42	998
20	14	14	0	14	14	56	1054
21	14	16	16	16	14	76	1130
22	14	16	16	16	14	76	1206
23	14	14	14	14	12	68	1274
24	12	12	12	12	12	60	1334
25	6	6	6	6	6	30	1364

According to the resource availability table, the scheduler forecasts the number of sorties, which the squadron has to accomplish in order to keep the timeline. For that reason, the same table is used for the notional schedule to evaluate the efficiency of the FSSBH and to analyze the outcomes of other mission order rules such as LNS and SNS.

SPs should be graduating from the B course by the end of the 25th week after the day they start flying (Table 6). A minimum of 1364 successful student sorties must be produced to achieve the timeline without any busts or other non-effective sorties. The non-effective missions due to busts, aircraft breakdowns or weather aborts cause the missions to be re flown. This 25-week timeline is computed by allowing slacks in the resource utilization to compensate for an 8% re-fly rate. In addition to these student sorties, approximately 1350 support sorties must be schedule and flown for 22 B course students to accomplish the missions according to the syllabus. In addition, 330 simulator flights must be scheduled for 22 SPs with IPs, further complicating the schedule.

Daily training requirements change as the SPs progress through the phases of the flight. As the senior class exits the program, a new class starts the B course to replace the old. While there is an overlap between two classes, this overlap does not impact the DFS.

4.2 Physical Structures and The Performance of the Software

Once the day's flights and simulator missions are realized (i.e. flown), the software has the necessary status inputs to build the DFS for the next day. The scheduler selects any order list on the spreadsheet and makes necessary adjustments to the MOL according to the status of the students and the resources. During this time, the software design and software performance can be measured.

The software design can be measured according to the interface environment and the flexibility of the software in the scheduling process. The software interface is user friendly and uncomplicated in that it mimics the sequence of activities that any scheduler at Oncel normally performs. The introduction screen provides two buttons, giving the option to go the main planning menu to select choices or to exit the program.

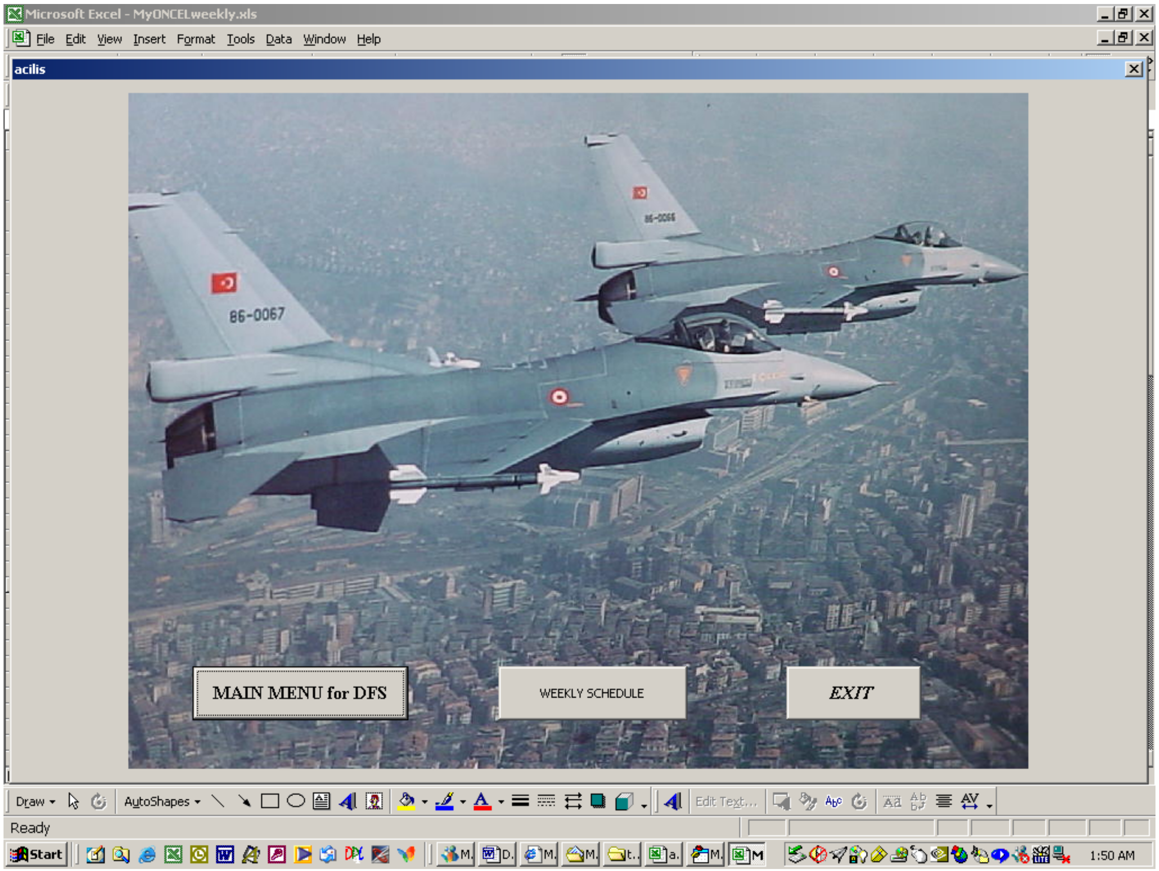

Microsoft Excel - MyONCELweekly.xls
File Edit View Insert Format Tools Data Window Help
acilis
86-0067
86-0066
MAIN MENU for DFS
WEEKLY SCHEDULE
EXIT
Draw AutoShapes Edit Text...
Ready
Start
1:50 AM

Figure 15. Start up Menu

The main menu provides a list of choices for the user to run separately so that he can interact with the schedule at any time. An option for the user to generate the DFS all at once is also provided. The list of choices provides the user the ability to update the mission order list, to clear the previous DFS and resource use tables, to sort the missions, to enter the data and the convert them into minutes, to assign pilots to duties, to generate the DFS either as a whole schedule or as a draft schedule and to complete the rest of the DFS after it is generated as draft. There is also an option to produce the simulator schedule if desired.

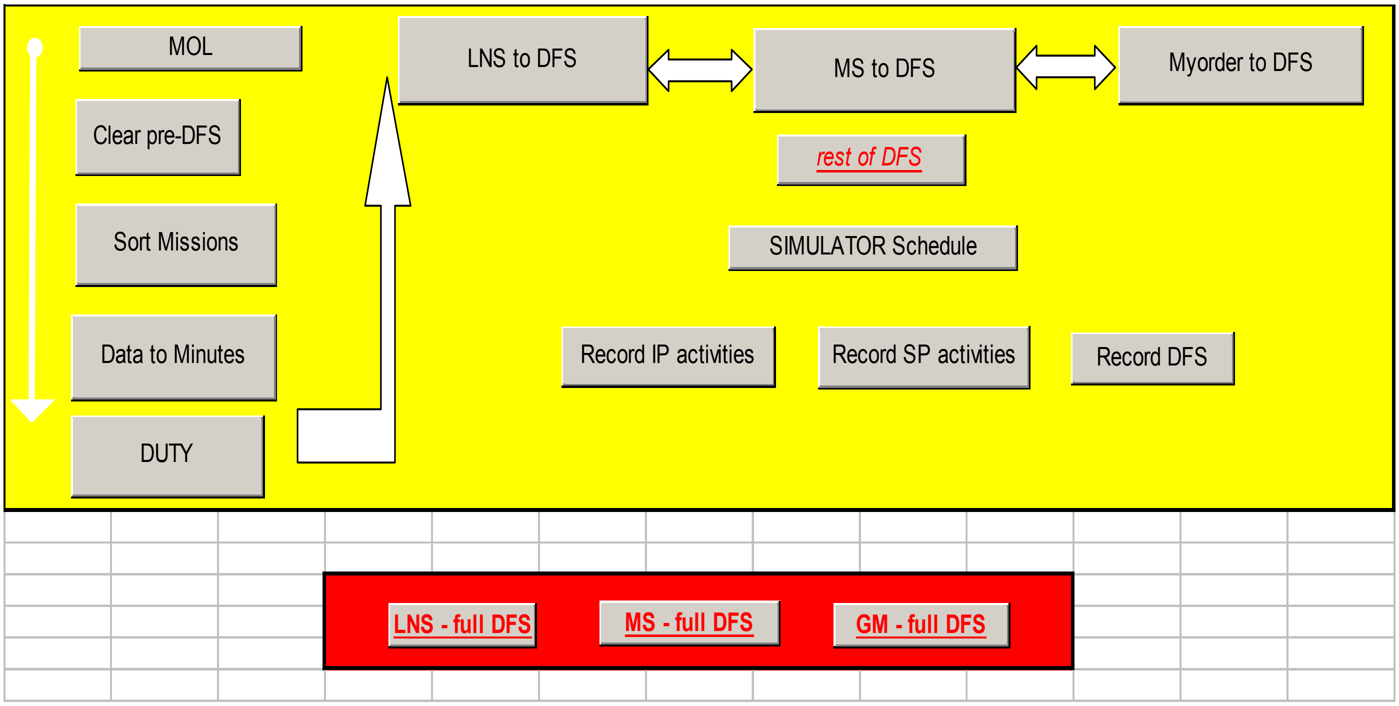


Figure 16. Main Menu

Post flight recording is included in the second part of the option list. After the DFS is executed, post flight recording allows flown sorties and simulators to be recorded on the mission lists. This updates the flight and simulator mission lists for the DFS. Additional options assist in recording the activities performed by a specific SP, IP and/or Bandit on an individual basis. Finally, another option records the DFS at the end of the day for data back up. These options are represented as buttons that have VBA codes written in the background of their respective spreadsheets. Pushing the appropriate button activates the codes assigned to it. These codes, as a collection, perform the activities necessary to generate the DFS or weekly schedule.

The Excel spreadsheet allows manual overrides of most functions to provide the scheduler with maximum flexibility. The scheduler may use one of the generated mission order lists or he has the option to set his own mission order list. One option allows the scheduler to generate the DFS as a draft schedule so that he can insert any mission into the schedule or he can shift or change the orders of the missions in the DFS. Another option allows the scheduler to assign the remaining resources to the altered draft schedule if changes have been made.

The scheduler can directly assign the IPs and Bandits. The rest of the schedule can be built afterwards or FTSSST can assign the people to duties according to the data entry table (Table 7). In a day there are three time windows, called "Go", available for the flight. The first go is from morning to noon, the second go is from noon to dusk and the third go is for night flying, which is from sunset to sunrise. An option available to the scheduler is to assign people to duties before or after the DFS is generated. The FTSSST reorders the pilots before assigning them to any activity so that a balance is maintained throughout the whole period – a period being either the 6-month flight program or the entire year.

Table 7. **Data Entry Table**

DATA ENTRY TABLE			CONVERSION TO MINUTES	
1st Go TO/Land time	10:00	12:50	600	770
2nd Go TO/Land time	13:45	16:20	825	980
3rd Go TO/Land time	20:00	23:30	1200	1410
RANGE	10:25	12:30	625	750
Area 1st Go Start/Finish T	10:00	12:40	600	760
Area 2nd Go Start/Finish T	14:00	16:00	840	960
Area 3rd Go Start/Finish T	20:00	23:30	1200	1410
RSU1	10:00	12:50	600	770
RSU2	141	NA	NA	NA
RSU3	142	NA	NA	NA
SOF1	142	NA	NA	NA
SOF2	14:20	17:00	860	1020
SOF3	141	NA	NA	NA
Area number (5or7)	5	5	5	5

After the daily flight data is entered into the data entry table, an option that converts the clock time into minutes is used just before the DFS is scheduled. Minutes, as a baseline period length, provide a very high resolution to the model. This resolution allows resource utilization to be more accurately portrayed. This is especially evident in the presence of scarce resources where this minute resolution allows the FTSSST to schedule sorties closer together in time versus models with lesser resolution. The overall affect is an increase in the number of sorties that can be scheduled through better implementation of the constraints in the model. If there is a takeoff –landing time conflict and/or landing-landing time conflict of the missions in the DFS, it is de-conflicted automatically by the software program on a minute basis as mentioned in Chapter 3, Section 3.3.

In addition to the tactical planning of the DFS, the FTSSST also provides a strategic planning tool. The FTSSST provides for the production of a multiple weekly schedule. After this option is chosen approximately 2600 missions can be produced in less than 2 minutes, the equivalent of the entire B course. The strategic planning tool of the FTSSST projects when a class will graduate based upon the given resources. Furthermore, FTSSST provides the opportunity for some sensitivity analysis. The squadron commander and the scheduler can determine their timeline and analyze the effects of any changes in the amount of the resources available for the squadron. At the end of the run, the FTSSST also provides how many sorties each IP and Bandit have to fly for the next 6-7 months to graduate the oncoming class subject to the number of the SPs and the resources available. The impact of using FTSSST for strategic planning is the ability to quantitatively defend requests for additional resources-either IPs, Bandits, aircraft, time and so on. This tool also produces the DFS without paying any attention to the current configuration of the aircraft. If there is a phase change from air-to-air to air-to-ground then this option provides a better weekly schedule because this option generates the weekly schedule without paying attention to aircraft configuration.

Once the data is entered, the scheduling algorithm can be run. The FTSSST is fast, in comparison to the current manual method, in generating the DFS, the weekly schedule, the simulator schedule, the course schedule and duty assignments. Running on a 866 MHz computer with 256 MB RAM, the FTSSST generates the DFS in less than 5 seconds. This DFS includes the Simulator schedule, course schedule and the duty assignments. The weekly schedule can be generated in less than 10 seconds on the same computer. As a result, the weekly schedule can be produced within seconds on the Thursday evening after the DFS for Friday is produced. This allows the scheduler and the squadron commander to see the next week's missions, anticipate problems, and take any necessary precautions such as changes in the leave policy for the IPs, simulator schedules with the other squadrons or course scheduling.

For any scheduler, making a draft flight schedule takes more than an hour sometimes two hours under the present manual approach. For the maintenance schedule, another 30 to 40 minutes are required to insert the aircraft information into the schedule. Then the schedule has to be written again in an Excel spreadsheet before being printed. If needed, the simulator schedule and the course schedule should be added to the DFS. The whole DFS production cycle, as it is currently performed, takes more than 2 hours with the assumption that there is no change in the status of any resource once it is made. If the status of any resource does change, then the DFS production cycle takes more than three hours to fully accomplish. During this time the schedulers and the maintenance personnel assigned to this task are busy. This time can be significantly reduced by the FTSSST. As previously mentioned, the FTSSST can generate an initial DFS in 5 seconds. Any change in the status of any resource adds another 5 seconds to the time necessary to produce a new schedule.

Weekly schedules work the same way as the DFS. There is no need to make any changes to any of the data used for the DFS. A "Weekly-Schedule" option produces a nominal schedule in less than 10 seconds that is ready to be printed. Contrast this to the half-day it takes to manually generate a weekly schedule. In all, the FTSSST generates a DFS and a weekly schedule in a relatively small amount of time compared to current methods.

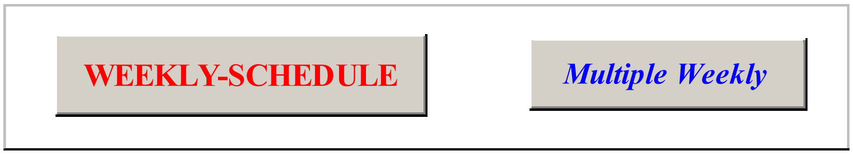


Figure 17. Weekly Schedule Options

4.3 Statistical Analysis and Performance of the MOLs and FSSBH

The notional training environment was used as training input to generate the DFS used for the analysis. The setup included 22 SPs and uses the data given in Table 5 and Table 7. The SPs started from the first phase of the flight program and are scheduled to fly until program completion. The three MOL rules being measured for the analysis are the Largest Number of Successors (LNS), Smallest Number of Successors (SNS) and the General Mission (GM) order with flight Scheduling shifting bottleneck heuristic (FSSBH) rules corresponding to the remaining mission numbers of a SP Furthest Behind the Training Schedule First, Furthest Ahead the Training Schedule First and FSSBH Determined Mission Order First.

The notional simulator and flight mission lists, shown in Appendix B, are used for the schedules. Data entries remained constant throughout the planning horizon for each MOL. The same construction heuristic is used for all of the MOLs, which are used to generate the DFS. The DFS is recorded for analysis after each run.

Table 8. Weekly Outcome Comparison of Sorties

weeks	Planned	LNS Order	SNS Order	GM order
1	50	68	70	69
2	40	70	70	71
3	50	67	66	61
4	50	55	72	71
5	54	78	81	74
6	60	80	78	80
7	60	84	73	88
8	60	85	71	86
9	60	84	81	88
10	60	85	80	88
11	36	80	74	81
12	60	68	94	83
13	60	70	87	74
14	60	70	66	91
15	60	70	63	81
16	62	74	50	70
17	74	79	46	67
18	42	69	35	41
19	56	28	35	
20	76		28	
21	76		22	
22	68		14	
23	60		8	
24	30			
total	1364	1364	1364	1364

The squadron scheduler was planning to graduate 22 SPs at the end of the 24th week after they started flying. The other three mission orders are used to generate the DFS with FTSSST and the results are displayed in Table 8.

The LNS first rule produced schedules that enabled the SPs to complete the B course by the end of the 19th week. On average, 14 sorties were scheduled per day. The LNS tended to keep SPs at about the same point within the flight program with 7 SPs graduating on the 91st day, and the remaining 15 pilots graduating on the 92nd and 93rd day. This indicates that the LNS performed as anticipated within the FTSSST.

The SNS First initially produced DFSs with high numbers of sorties. This high sortie generation continued until the end of 13th week. At this point, the number of sorties produced has dramatically decreased. One reason for this drop is that 8 SPs completed the program by the end of 13th week. This reduction in the sortie production rate continues each week as 2-4 SPs graduate every 5-6 days until all SPs have completed the program (the end of week 23). Again, these results indicate that the MS rule performed as anticipated within the FTSSST.

The GM order rule is used for the FSSBH. It is a combination of LNS First rule and SNS rule. Recall from Chapter 3 section 3.3.1 that, the GM order rule implements the heuristic to build up the mission order list according to the available squadron resources to shift the bottleneck. Additionally, the GM order heuristic makes some alterations to the MOL, according to the remaining number of missions of the SPs.

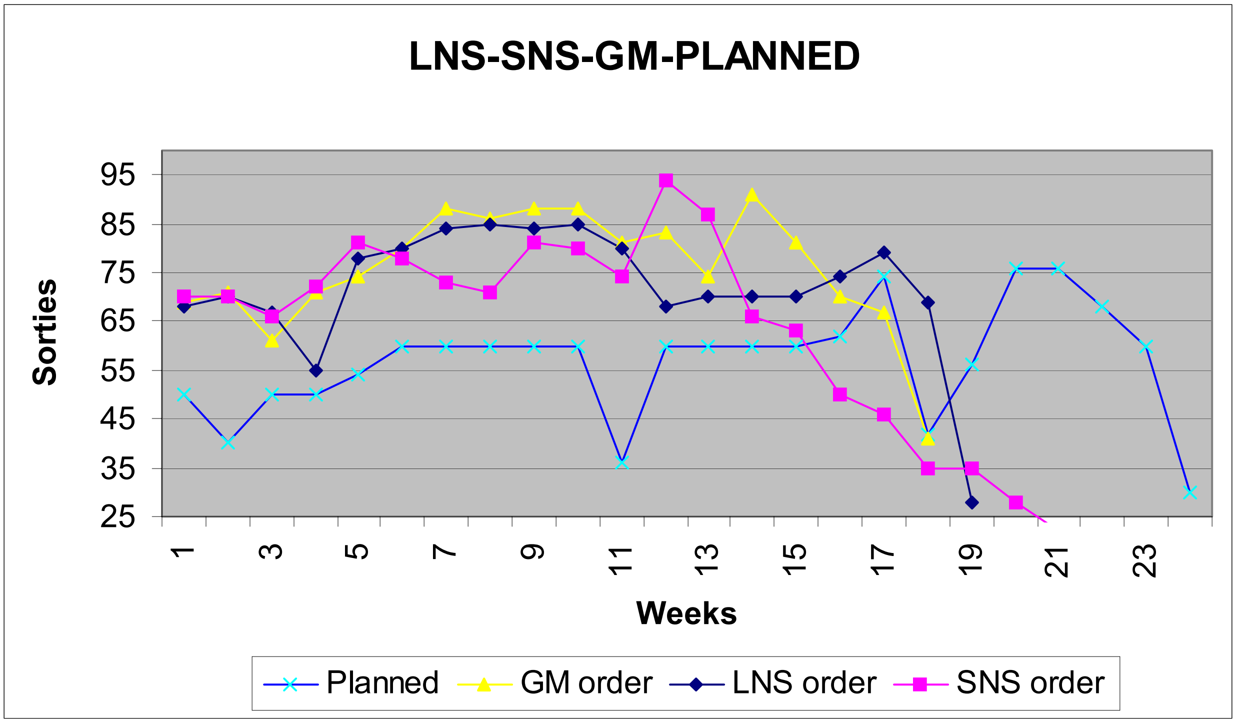


Figure 18. LNS rule vs. SNS rule vs. GM order vs. Planned sorties

The GM rule with the FSSBH produced a schedule enabled all of the SPs to complete the program by the end of the 18th week. On average, 15.3 sorties per day were generated. The GM rule balanced the demand for scarce resources across the program. This was accomplished by allowing some students to push ahead in the program schedule. Again, the results indicate that the GM rule performed as anticipated.

As the LNS rule, FSSBH also provides a very balanced workload to the squadron over the weeks. Unlike the other rules, SNS has a very poor workload balance.

The FTSSST also provides a balanced workload to the IPs and Bandits on the individual basis. This balance is independent from the priority rule used to generate the

DFS. In Oncel, IPs and Bandits perform various activities with duties and flights as the primary activities.

Table 9. Flight and Simulator Statistics

Flight Statistics			Simulator Statistics	
Mean	65.83333		Mean	25.46667
Standard Error	0.179932		Standard Error	0.092641
Median	66		Median	25
Mode	66		Mode	25
Standard Deviation	0.985527		Standard Deviation	0.507416
Sample Variance	0.971264		Sample Variance	0.257471
Kurtosis	-0.17029		Kurtosis	-2.12691
Skewness	-0.80197		Skewness	0.140769
Range	3		Range	1
Minimum	64		Minimum	25
Maximum	67		Maximum	26
Sum	1975		Sum	764
Count	30		Count	30

Recall from Chapter 3 Section 3.3 that, the FTSSST accumulates the performed activities of each individual and assigns him to the next job according to his past performed actions. If the scheduler has to manually assign somebody to a mission or duty, this recorded data provides a support for his decision.

Descriptive statistics for the flight and the simulator missions as performed by the IPs and the Bandits are provided in Table 9. For a period of a B Course, the FTSSST had scheduled any IP or Bandit in Oncel between 64 and 67 times for the flight missions and between 25 to 26 times for the simulator mission. The sum of flight sorties, 1975, given in Table 9, is the sum of support sorties and instructor sorties. The sample variance is very small (<1) for both the flights and the simulator as seen in Table 9. These statistics implies that the FTSSST is providing a very good balance, which is desirable by every pilot, among the members of the squadron.

In addition to the flight and simulator balance, the FTSSST also maintained the number of additional duties performed by the IPs and Bandits and balanced these duties as well. According to military rank, the personnel in the squadron are divided in two groups.

Table 10. SOF and RSU Statistics

SOF Statistics			RSU Statisitics	
Mean	6.714286		Mean	5.75
Standard Error	0.125294		Standard Error	0.111803
Median	7		Median	6
Mode	7		Mode	6
Standard Deviation	0.468807		Standard Deviation	0.447214
Sample Variance	0.21978		Sample Variance	0.2
Kurtosis	-1.03409		Kurtosis	-0.43956
Skewness	-1.06654		Skewness	-1.27775
Range	1		Range	1
Minimum	6		Minimum	5
Maximum	7		Maximum	6
Sum	94		Sum	92
Count	14		Count	16

The first half performs SOF duty and the second half performs the RSU duty. Each pilot in the first group has performed 6 or 7 SOF duties. In the second group, each pilot has performed RSU duty either 5 or 6 times.

The FTSSST has achieved the objective of maintaining the balance among the pilots on the individual basis as shown by the descriptive statistics in Table 9 and Table 10.

4.3.1 Priority Rules Comparisons

GM order achieved the shortest program completion time and maintained a balanced workload throughout the DFSs. The LNS rule achieved nearly the same program completion time as the GM rule and also maintained a balanced workload. The

SNS rule managed to graduate students sooner than the other two rules, but did not graduate every student until much later than the LNS and GM rules. (A longer C_{max}, in scheduling terms).

The LNS First rule had very close outcomes compared to the GM order. Overall, GM order had finished the schedule a week prior to the LNS First rule. In the flight scheduling environment even a day is very important; a week advantage is very valuable.

If the three rules are compared to each other, GM order had the best throughput rate per period, LNS First rule is the second best one and the SNS First rule had very poor results by itself. The SNS First Rule plays a very important role in making the GM order, so that it has still a very good implementation in the FTSSST.

4.4 ILP solutions vs. the FTSSST solutions

Silver, *et al*, suggests that a good heuristic should posses the following properties [Silver, *et al*, p.153-162, 1980]:

1. Realistic computational effort to obtain the solution.
2. The solution should be close to the optimum on the average, i.e., good performance on the average is desired
3. The chance of a very poor solution (i.e., far from the optimum) should be low.
4. The heuristic should be as simple as possible for the user to understand, preferable understandable in intuitive terms, particularly if it is to be used manually. Carefully prepared documentation should help the end user.

This section concentrated on the measurement of quality primarily in terms of properties 2 and 3.

> One would like to be able to compare the heuristic solution with the best possible over a large number of problem instances. Usually this is not possible, in that, as mentioned in chapter 2, a major reason for using a heuristic procedure in the first place is that it may be impossible or prohibitive from a computational standpoint to obtain the optimal solution. Also it may be necessary to concentrate on small scale problems (of smaller size than at least some of the instances of interest) to reduce the computational effort to a reasonable level. [Silver, *et al*, p.153-162, 1980].

"An alternative approach is to relax the problem so that a solution can be evaluated that is as good as the optimum solution if the optimum solution cannot be found" [Silver, *et al*, p.153-162, EJOR]. This will provide an upper bound on the optimum solution. This is a one-way test; the optimal value must lie between the value of the heuristic solution and the bound. If the value of the heuristic is very close to the bound then it must be very close to the optimum solution value.

The most common way of a relaxing a problem is to ignore one or more constraints. Recall from Chapter 2 Figure 8 that the Complete MMGRCPSP for DFS is given. This provides mathematical basis for the scheduling problem in Oncel.

The relaxation of this model included the following:

- Takeoff times are separated
- Resolution is lowered by aggregating minutes into 10-minute periods
- Aggregation of time periods caused values to be rounded down

This relaxation provides an upper bound to the problem. The FTSSST can go down to a resolution of a minute while generating the DFS so that upper bound could be provided. A period stands for 10 minutes. The same data entry table was prepared for ILP and the variables were generated by a code written in VBA. 10 problem instances are created to reflect the possible MOLs that the FTSSST may encounter in the DFS production cycle through the program (Appendix C). A mission list is prepared on the spreadsheet and variables are defined according the available periods and modes. If the variables are more than 200 then some of the students were blocked to reduce the number of variables to not exceed the capacity of the standard Solver included with Excel. Each problem was solved in two parts. At first, the Morning Go was scheduled. If a SP was scheduled then his name was removed from the mission list. The Afternoon Go was then solved according to the remaining missions from the first go with the Excel solver.

Table 11. Mission Resource Use Vector and Availability

	FLIGHT								souce1	souce 2
Resources	**brief**	**gnd ops**	**ingress**	**area work**	**egress**	**landing**	**shutdown**	**debrief**	**availability**	**availability**
IP	1	1	1	1	1	1	1	1	30	30
SP	1	1	1	1	1	1	1	0	30	30
D	0	1	1	1	1	1	1	0	8	8
C	0	1	1	1	1	1	1	0	26	26
A	0	0	0	1	0	0	0	0	5	5
R	0	0	0	1	0	0	0	0	1	1
S	0	0	0	1	0	0	0	0	1	1

After the variables were defined, based on the Table 11, the resource matrices had been built by a code written in VBA.

Table 12. Mode Vectors for the Missions

	Mis. Name	TR-1	TR-2	TR-3	TR-4	TR-5	TR-6	TR-7	BIF	**TR-8**	INT-1
	Mis. No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Resources	**Mode**	**1**	**1**	**1**	**1**	**2**	**1**	**3**	**4**	**1**	**2**
IP		1	1	1	1	1	1	1	1	1	1
SP		0	0	0	0	1	0	0	0	0	1
D		1	1	1	1	1	1	0	1	1	1
C		0	0	0	0	1	0	2	0	0	1
A		1	1	1	1	1	1	1	0	1	1
R		0	0	0	0	0	0	0	0	0	0
	Duration	5	4	4	4	4	3	3	5	5	6

Table 13. Total Mission Vector

	brief	gnd ops	ingress	area work	egress	landing	shutdown	debrief	total
mission time	6	6	1	0	2	2	4	6	27
simtime	6	0	0	6	0	0	0	6	

Total mission time consists of the sum of the periods shown in Table 13 and the mission duration shown in the last row of Table 12. This implementation allowed missions to have different processing times based on their durations. It also reduced the duplication of effort to calculate the mission times of fixed processing times such as briefing, shutdown and so forth.

Table 14. ILP vs. FTSSST and the Solution Statistics

Prob. No	ILP	FTSSST	Difference	t-Test: Paired Two Sample for Means		
1	16	15	1			
2	15	14	1		Variable 1	Variable 2
3	16	15	1	Mean	16.5	15.8
4	12	12	0	Variance	4.72	4.4
5	17	15	2	Observations	10	10
6	16	16	0	Pearson Correlation	0.95	
7	19	18	1	Hypothesized Mean Difference	0	
8	19	19	0	df	9	
9	16	16	0	t Stat	3.28	
10	19	18	1	P(T<=t) one-tail	0.005	
st dev	2.17	2.10	0.67	t Critical one-tail	1.83	
mean	16.5	15.8	0.7	P(T<=t) two-tail	0.01	
				t Critical two-tail	2.26	

The relaxed ILP solutions and the FTSSST solution are very given on the left side of the Table 14. The FTSSST had results very close to the upper bound in all of the problem sets.

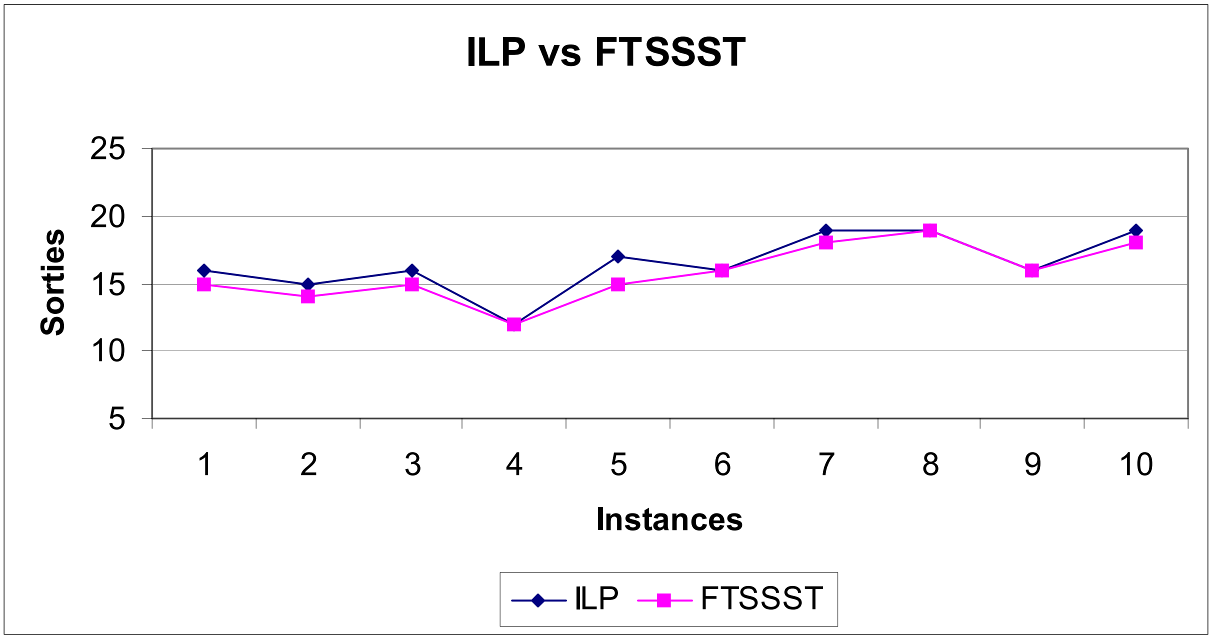


Figure 19. ILP vs. the FTSSST solutions

The FTSSST had 4 optimum results out of 10 problem instances. In 5 instances it had produced only one mission less than the upper bound and in one instance it produced 2 less than the upper bound.

Based on the paired t test shown in Table 14, one can conclude that with 95 percent confidence interval, the ILP solutions makes an upper bound for this particular scheduling problem and can physically determine that the FTSSST solutions are very close to the upper bound.

4.4 Summary

The goal of this research has been to present the squadron schedulers with an automated scheduling capability. The complete MMGRCPSP for DFS is well suited for the scheduling heuristic. The methods presented in this thesis allow schedulers to obtain extremely fast, close to upper bound solutions by using the initial construction heuristic.

In addition to the tactical solutions it provides, the FTSSST can be used strategically. By generating the DFS and the weekly schedules according to the current syllabus, the FTSSST frees the scheduler's time to attend to the details and variations that cannot be programmed. Chapter 5 presents the contributions of this research, recommends future works as follow on.

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

This chapter reviews the importance of this research as well as the major issues covered in this research. The key points are reviewed, the significant contributions are outlined, and recommendations for future researches are suggested.

5.1 Research

The research for this thesis pursued three primary lines of investigation. First, the nature and the scope of the scheduling problem were defined by examining the current Oncel scheduling process for the dynamics involved in producing a daily and weekly flight schedule. Second, the research investigated the concepts of the scheduling theory to find parallels to the flight-scheduling problem. The third line of investigation delved into the heuristics to develop an application that could be used by schedulers at Oncel.

The scheduling process at Oncel displays similar characteristics to manufacturing systems. Activities, such as simulator missions, flight missions or additional duties, have processing times and due dates. Resources, such as aircraft, IPs or Bandits, are renewable and non-renewable within the DFS; hence, they are used in the schedule accordingly. Scheduling in a training environment is dynamic because of bad weather, aircraft breakdowns and pilot sick calls that can occur at any time. The requirements imposed by regulations and rules must be enforced while generating the DFS. The scheduling directly influences every pilot's life in the squadron. Therefore, a balanced workload should be maintained among the IPs and Bandits while a good flight flow is provided for the SPs. The weekly schedule provides a good indication of the next week's expected missions so that the squadron personnel and other support units can take actions to avoid training interruption.

5.2 Contributions

Several important contributions were provided through this research. The first contribution is a fast heuristic approach for the scheduling problem that incorporates aspects of resource utilization and mission ordering. The construction heuristic performs with any given mission list and daily sortie production rate is very close to the upper bound solution provided by the relaxed ILP formulation. The construction heuristic has found the optimal in 4 instances out of 10. Given the dynamics of flight scheduling, the speed of this heuristic is invaluable. Overall, the construction heuristic performed well for any MOL. The software developed here reads from a Microsoft Access database. Options on the spreadsheet allow some data to be updated automatically. Flight and simulator mission lists can be updated and/or changed easily using the MS Access database provided.

Another contribution of this research is the shifting bottleneck heuristic. The FSSBH yielded good solutions through the program in comparison to the other priority rules. The FSSBH also has the flexibility to be changed by the end user if needed due to resource availability, i.e. more aircraft, longer time windows, or a change in the syllabus.

The third contribution of this research is the workload balance provided through the program. At the end of the program, the IPs and Bandits fly approximately the same number of simulator and flight missions and they are also scheduled for the same number of additional duties. In the long run, this should provide positive motivation for the squadron personnel as well increase the quality of the instruction as.

In addition to the tactical contributions of this research, it also provides a strategic contribution. The FTSSST can also schedule the whole B course in less then 5 minutes. The following results are reported by the FTSSST; the number of flight missions flown per IP and Bandit, SOF and RSU duties performed per person, how long it takes to graduate that particular group of students and so on. This option may provide a better future forecast and planning to the decision makers.

5.3 Recommendations for Future Work

The research contained within this thesis may be extended in a number of directions. Some of these are:

1. The FSSBH can be improved by incorporating a more in depth study of the resources at Oncel.
2. An attrition model can be applied to the schedule once the DFS is produced for the post sortie analysis.
3. Tabu Search or other heuristic approaches can be implemented along with the FSSBH.
4. Rescheduling can be implemented with a goal-programming model.
5. The relaxed ILP solution can be found for all instances of the DFS using resolutions at a minute at a minute-by-minute basis.
6. Software and database relation can be improved so that at the end of each flight day the data is stored automatically.
7. Resource availability data can be gathered and implanted in to the model with their distributions, so that the software can be used for planning strategically.

8. The software can be extended to produce schedules with the predetermined takeoff times as well

5.4 Summary

A method for finding fast and good solutions to the squadron-scheduling problem was developed during the course of this work. The method can be applied to the other training squadrons as well. The application has the aspects of tactical and strategical implementation.

Appendix A. List of Abbreviations

AA	Air to Air
AAR	Air-to-Air Refueling
ACM	Air Combat Maneuvering
AFB	Air Force Base
AG	Air to Ground
AREC	Armed Reconnaissance
BFM	Basic Fighter Maneuvering
BI	Basic Intercept
BTR	Basic Training
COB	Close of Business
DACT	Different Type Aircraft Combat Maneuvering
DFS	Daily Flight Schedule
EJOR	European Journal of Operation Research
FSSBH	Flight Scheduling Shifting Bottleneck Heuristic
FTSSST	Flight Training Squadron Scheduling Support Tool
GM	General Mission Order
GUI	Graphical User Interface
INT	Intercept
IP	Instructor Pilot
IQ	Instrument Qualification

LFJ	Least Flexible Job
LFM	Least Flexible Machine
LNS	Largest Number of Successors
MIP	Mixed Integer Programming
MMGRCPSP	Multi-Modal Resource Constrained Project Scheduling Problem
MOL	Mission Order List
NI	Night Intercept
NTR	Night Training
OOP	Object Oriented Programming
RSU	Runway Supervisory Unit
SA	Surface Attack
SAT	Tactical Surface Attack
SBH	Shifting Bottleneck Heuristic
SOF	Supervisory of Flight
SNS	Smallest Number of Successors
SP	Student Pilot
STA	Surface Tactical Attack
TQ	Tactical Qualification
TUAF	Turkish Air Force
VBA	Visual Basic for Applications

Appendix B. Problem Sets

For this research the following 10 problem instances are generated to find an upper bound for the construction heuristic. The instances are chosen according to the possible conditions that can be encountered in the DFS production cycle. The relaxed ILP formulation of the problem is shown in Figure 17.

Parameters:

A the set of all activities (related to student *m*)

d the index of terminal activity

e_{im} the earliest completion time for activity *i* in mode *m*

l_{im} the latest completion time for activity *i* in mode *m*

τ_{im} the duration of activity *i* in mode *m*

r_{imk} the amount of resource *k* required by activity when being executed in mode *m*

R_k the per period availability of resource *k*

Variables:

x_{imt} 1 if activity starts in period t and is executed in mode m

Maximize

$$\sum_{m \in M_d} \sum_{i \in A} x_{imt} \qquad (1)$$

Subject to:

$$\sum_{i \in A} \sum_{m \in M_i} \sum_{t=\tau_{im}+1}^{j} r_{imk}\, x_{imt} \le R_k \qquad \forall\, k \in K \qquad (2)$$

$$\sum_{m \in M_i} \sum_{t=e_{im}}^{l_{im}} x_{imt} \le 1 \qquad \forall\, i \in A \qquad (3)$$

$$\sum_{t=e_{im}}^{l_{im}} x_{imt} \le 1 \qquad \forall\, i \in A \qquad (4)$$

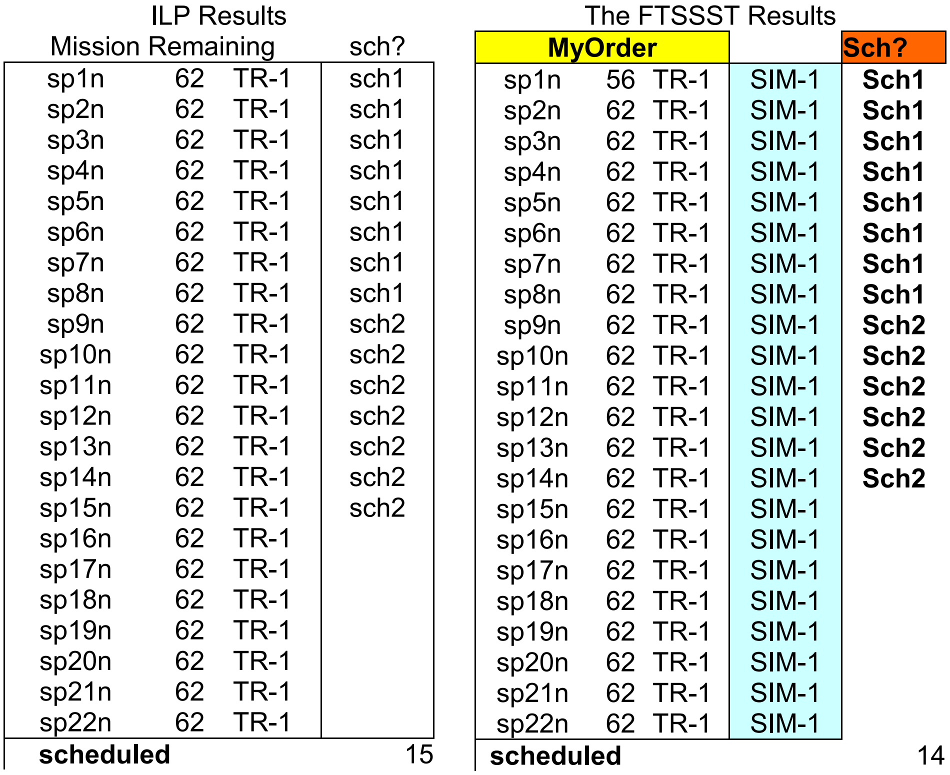
$$x_{imt} \in \{0,1\} \qquad \forall\, (i,m,t)$$

Figure 20. Relaxed ILP formulation for DFS

The instances and the results are as follows:

1^{st} instance; the first instance mimic the start of the B course. All of the SPs are ready to fly their first mission in the program. Sch1 means the mission scheduled in the first go and sch2 means the mission is scheduled in the second go, that is, the afternoon.

Table 15. 1^{st} Instance and its Results



ILP Results

Mission Remaining			sch?
sp1n	62	TR-1	sch1
sp2n	62	TR-1	sch1
sp3n	62	TR-1	sch1
sp4n	62	TR-1	sch1
sp5n	62	TR-1	sch1
sp6n	62	TR-1	sch1
sp7n	62	TR-1	sch1
sp8n	62	TR-1	sch1
sp9n	62	TR-1	sch2
sp10n	62	TR-1	sch2
sp11n	62	TR-1	sch2
sp12n	62	TR-1	sch2
sp13n	62	TR-1	sch2
sp14n	62	TR-1	sch2
sp15n	62	TR-1	sch2
sp16n	62	TR-1	
sp17n	62	TR-1	
sp18n	62	TR-1	
sp19n	62	TR-1	
sp20n	62	TR-1	
sp21n	62	TR-1	
sp22n	62	TR-1	
scheduled			15

The FTSSST Results

MyOrder				**Sch?**
sp1n	56	TR-1	SIM-1	**Sch1**
sp2n	62	TR-1	SIM-1	**Sch1**
sp3n	62	TR-1	SIM-1	**Sch1**
sp4n	62	TR-1	SIM-1	**Sch1**
sp5n	62	TR-1	SIM-1	**Sch1**
sp6n	62	TR-1	SIM-1	**Sch1**
sp7n	62	TR-1	SIM-1	**Sch1**
sp8n	62	TR-1	SIM-1	**Sch1**
sp9n	62	TR-1	SIM-1	**Sch2**
sp10n	62	TR-1	SIM-1	**Sch2**
sp11n	62	TR-1	SIM-1	**Sch2**
sp12n	62	TR-1	SIM-1	**Sch2**
sp13n	62	TR-1	SIM-1	**Sch2**
sp14n	62	TR-1	SIM-1	**Sch2**
sp15n	62	TR-1	SIM-1	
sp16n	62	TR-1	SIM-1	
sp17n	62	TR-1	SIM-1	
sp18n	62	TR-1	SIM-1	
sp19n	62	TR-1	SIM-1	
sp20n	62	TR-1	SIM-1	
sp21n	62	TR-1	SIM-1	
sp22n	62	TR-1	SIM-1	
scheduled				14

2^{nd} instance; in this problem, the students are scattered in the first phase of the flight. The mission order is given arbitrarily for the FTSSST.

Table 16. 2nd and 3rd Instances and Results

ILP Results			
mission	remaining		sch?
sp1n	61	TR-2	sch1
sp2n	60	TR-3	sch1
sp3n	60	TR-3	sch1
sp4n	60	TR-3	sch1
sp5n	59	TR-4	sch1
sp6n	59	TR-4	sch1
sp7n	59	TR-4	sch1
sp8n	59	TR-4	sch1
sp9n	59	TR-4	sch2
sp10n	59	TR-4	sch2
sp11n	59	TR-4	sch2
sp12n	60	TR-3	sch2
sp13n	60	TR-3	sch2
sp14n	60	TR-3	sch2
sp15n	62	TR-1	sch2
sp16n	61	TR-2	sch2
sp17n	61	TR-2	
sp18n	62	TR-1	
sp19n	61	TR-2	
sp20n	61	TR-2	
sp21n	62	TR-1	
sp22n	62	TR-1	
scheduled			16

The FTSSST Results				
MyOrder				**Sch?**
sp1n	61	TR-2	SIM-1	**Sch1**
sp2n	60	TR-3	SIM-1	**Sch1**
sp3n	60	TR-3	SIM-3	**Sch1**
sp4n	60	TR-3	SIM-3	**Sch1**
sp5n	59	TR-4	SIM-1	**Sch1**
sp6n	59	TR-4	SIM-1	**Sch1**
sp7n	59	TR-4	SIM-3	**Sch1**
sp8n	59	TR-4	SIM-1	**Sch1**
sp9n	59	TR-4	SIM-3	**Sch2**
sp10n	59	TR-4	SIM-1	**Sch2**
sp11n	59	TR-4	SIM-3	**Sch2**
sp12n	60	TR-3	SIM-1	**Sch2**
sp13n	60	TR-3	SIM-1	**Sch2**
sp14n	60	TR-3	SIM-1	**Sch2**
sp15n	62	TR-1	SIM-3	**Sch2**
sp16n	61	TR-2	SIM-1	
sp17n	61	TR-2	SIM-3	
sp18n	62	TR-1	SIM-1	
sp19n	61	TR-2	SIM-1	
sp20n	61	TR-2	SIM-1	
sp21n	62	TR-1	SIM-1	
sp22n	62	TR-1	SIM-1	
scheduled				15

ILP Results			
mission	remaining		sch?
sp1n	61	TR-2	sch1
sp2n	60	TR-3	sch1
sp3n	60	TR-3	sch1
sp4n	60	TR-3	sch1
sp5n	59	TR-4	sch1
sp6n	59	TR-4	sch1
sp7n	59	TR-4	sch1
sp8n	59	TR-4	sch1
sp9n	50	INT-4	sch1
sp10n	50	INT-4	sch2
sp11n	50	INT-4	sch2
sp12n	52	INT-2	sch2
sp13n	52	INT-2	sch2
sp14n	51	INT-3	sch2
sp15n	52	INT-2	
sp16n	52	INT-2	
sp17n	52	INT-2	
sp18n	52	INT-2	
sp19n	50	INT-4	
sp20n	61	TR-2	sch2
sp21n	62	TR-1	
sp22n	62	TR-1	sch2
scheduled			16

The FTSSST Results				
MyOrder				**Sch?**
sp1n	61	TR-2	SIM-1	**Sch1**
sp2n	60	TR-3	SIM-1	**Sch1**
sp3n	60	TR-3	SIM-3	**Sch1**
sp4n	60	TR-3	SIM-3	**Sch1**
sp5n	59	TR-4	SIM-1	**Sch1**
sp6n	59	TR-4	SIM-1	**Sch1**
sp7n	59	TR-4	SIM-3	**Sch1**
sp8n	59	TR-4	SIM-1	**Sch1**
sp9n	50	INT-4	SIM-3	**Sch2**
sp10n	50	INT-4	SIM-1	**Sch2**
sp11n	50	INT-4	SIM-3	**Sch2**
sp12n	52	INT-2	SIM-1	**Sch2**
sp13n	52	INT-2	SIM-1	**Sch2**
sp14n	51	INT-3	SIM-1	
sp15n	52	INT-2	SIM-3	
sp16n	52	INT-2	SIM-1	
sp17n	52	INT-2	SIM-3	
sp18n	52	INT-2	SIM-1	
sp19n	50	INT-4	SIM-1	
sp20n	61	TR-2	SIM-1	**Sch2**
sp21n	62	TR-1	SIM-1	**Sch2**
sp22n	62	TR-1	SIM-1	
scheduled				15

3rd instance; in this problem, some of the students are through the first phase of the B course and ready to start the INT phase.

In the 4th instance the missions are chosen from the INT missions.

Table 17. 4th Instance and its Results

ILP Results

mission remaining			sch?
sp1n	53	INT-1	sch1
sp2n	53	INT-1	sch1
sp3n	53	INT-1	sch1
sp4n	53	INT-1	sch1
sp5n	53	INT-1	sch1
sp6n	53	INT-1	sch1
sp7n	53	INT-1	sch2
sp8n	53	INT-1	sch2
sp9n	53	INT-1	sch2
sp10n	53	INT-1	sch2
sp11n	53	INT-1	sch2
sp12n	53	INT-1	
sp13n	53	INT-1	
sp14n	53	INT-1	
sp15n	53	INT-1	
sp16n	53	INT-1	
sp17n	53	INT-1	
sp18n	53	INT-1	
sp19n	53	INT-1	
sp20n	53	INT-1	
sp21n	53	INT-1	
sp22n	53	INT-1	sch1
scheduled			12

The FTSSST Results

MyOrder				**Sch?**
sp1n	53	INT-1	SIM-1	**Sch1**
sp2n	53	INT-1	SIM-1	**Sch1**
sp3n	53	INT-1	SIM-3	**Sch1**
sp4n	53	INT-1	SIM-3	**Sch1**
sp5n	53	INT-1	SIM-1	**Sch1**
sp6n	53	INT-1	SIM-1	**Sch1**
sp7n	53	INT-1	SIM-3	**Sch2**
sp8n	53	INT-1	SIM-1	**Sch2**
sp9n	53	INT-1	SIM-3	**Sch2**
sp10n	53	INT-1	SIM-1	**Sch2**
sp11n	53	INT-1	SIM-3	**Sch2**
sp12n	53	INT-1	SIM-1	**Sch2**
sp13n	53	INT-1	SIM-1	
sp14n	53	INT-1	SIM-1	
sp15n	53	INT-1	SIM-3	
sp16n	53	INT-1	SIM-1	
sp17n	53	INT-1	SIM-3	
sp18n	53	INT-1	SIM-1	
sp19n	53	INT-1	SIM-1	
sp20n	53	INT-1	SIM-1	
sp21n	53	INT-1	SIM-1	
sp22n	53	INT-1	SIM-1	
scheduled				12

In the 5th, 6th, 7th and 8th instances; the missions are chosen from the INT-BFM, BFM and the BFM-ACM mission combination.

Table 18. 5th and 6th Instances and Results

ILP Results

mission remaining			sch?
sp1n	44	BFM-2	sch1
sp2n	42	BFM-4	sch1
sp3n	43	BFM-3	sch1
sp4n	44	BFM-2	sch1
sp5n	39	BFM-7	sch1
sp6n	45	BFM-1	sch1
sp7n	41	BFM-5	sch1
sp8n	45	BFM-1	sch1
sp9n	40	BFM-6	sch1
sp10n	52	INT-2	sch1
sp11n	53	INT-1	sch2
sp12n	52	INT-2	sch2
sp13n	50	INT-4	sch2
sp14n	41	BFM-5	sch2
sp15n	50	INT-4	sch2
sp16n	52	INT-2	sch2
sp17n	51	INT-3	
sp18n	51	INT-3	
sp19n	37	BFM-9	sch2
sp20n	52	INT-2	
sp21n	53	INT-1	
sp22n	53	INT-1	
scheduled			17

The FTSSST Results

MyOrder				Sch?
sp1n	44	BFM-02	SIM-1	**Sch1**
sp2n	42	BFM-04	SIM-1	**Sch1**
sp3n	43	BFM-03	SIM-3	**Sch1**
sp4n	44	BFM-02	SIM-3	**Sch1**
sp5n	39	BFM-07	SIM-1	**Sch1**
sp6n	45	BFM-01	SIM-1	**Sch1**
sp7n	41	BFM-05	SIM-3	**Sch1**
sp8n	45	BFM-01	SIM-1	**Sch1**
sp9n	40	BFM-06	SIM-3	**Sch1**
sp10n	52	INT-2	SIM-1	**Sch2**
sp11n	53	INT-1	SIM-3	**Sch2**
sp12n	52	INT-2	SIM-1	**Sch2**
sp13n	50	INT-4	SIM-1	**Sch2**
sp14n	41	BFM-05	SIM-1	**Sch2**
sp15n	50	INT-4	SIM-3	
sp16n	52	INT-2	SIM-1	
sp17n	51	INT-3	SIM-3	
sp18n	51	INT-3	SIM-1	
sp19n	37	BFM-09	SIM-1	**Sch2**
sp20n	52	INT-2	SIM-1	
sp21n	53	INT-1	SIM-1	
sp22n	53	INT-1	SIM-1	
scheduled				15

ILP Results

mission remaining			sch?
sp1n	45	BFM-1	sch1
sp2n	45	BFM-1	sch1
sp3n	45	BFM-1	sch1
sp4n	45	BFM-1	sch1
sp5n	45	BFM-1	sch1
sp6n	45	BFM-1	sch1
sp7n	45	BFM-1	sch1
sp8n	45	BFM-1	sch1
sp9n	45	BFM-1	sch2
sp10n	45	BFM-1	sch2
sp11n	45	BFM-1	sch2
sp12n	45	BFM-1	sch2
sp13n	45	BFM-1	sch2
sp14n	45	BFM-1	sch2
sp15n	45	BFM-1	sch2
sp16n	45	BFM-1	sch2
sp17n	45	BFM-1	
sp18n	45	BFM-1	
sp19n	45	BFM-1	
sp20n	45	BFM-1	
sp21n	45	BFM-1	
sp22n	45	BFM-1	
scheduled			16

The FTSSST Results

MyOrder				Sch?
sp1n	45	BFM-01	SIM-1	**Sch1**
sp2n	45	BFM-01	SIM-1	**Sch1**
sp3n	45	BFM-01	SIM-3	**Sch1**
sp4n	45	BFM-01	SIM-3	**Sch1**
sp5n	45	BFM-01	SIM-1	**Sch1**
sp6n	45	BFM-01	SIM-1	**Sch1**
sp7n	45	BFM-01	SIM-3	**Sch1**
sp8n	45	BFM-01	SIM-1	**Sch1**
sp9n	45	BFM-01	SIM-3	**Sch2**
sp10n	45	BFM-01	SIM-1	**Sch2**
sp11n	45	BFM-01	SIM-3	**Sch2**
sp12n	45	BFM-01	SIM-1	**Sch2**
sp13n	45	BFM-01	SIM-1	**Sch2**
sp14n	45	BFM-01	SIM-1	**Sch2**
sp15n	45	BFM-01	SIM-3	**Sch2**
sp16n	45	BFM-01	SIM-1	**Sch2**
sp17n	45	BFM-01	SIM-3	
sp18n	45	BFM-01	SIM-1	
sp19n	45	BFM-01	SIM-1	
sp20n	45	BFM-01	SIM-1	
sp21n	45	BFM-01	SIM-1	
sp22n	45	BFM-01	SIM-1	
scheduled				16

Table 19. 7th and 8th Instances and Results

ILP Results			
mission remaining			sch?
sp1n	39	BFM-7	sch1
sp2n	40	BFM-6	sch1
sp3n	40	BFM-6	sch1
sp4n	41	BFM-5	sch1
sp5n	41	BFM-5	sch1
sp6n	42	BFM-4	sch1
sp7n	42	BFM-4	sch1
sp8n	42	BFM-4	sch1
sp9n	42	BFM-4	sch1
sp10n	43	BFM-3	sch1
sp11n	43	BFM-3	sch2
sp12n	43	BFM-3	sch2
sp13n	43	BFM-3	sch2
sp14n	43	BFM-3	sch2
sp15n	43	BFM-3	sch2
sp16n	44	BFM-2	sch2
sp17n	44	BFM-2	sch2
sp18n	44	BFM-2	sch2
sp19n	44	BFM-2	sch2
sp20n	44	BFM-2	
sp21n	44	BFM-2	
sp22n	44	BFM-2	
scheduled			19

The FTSSST Results				
MyOrder				**Sch?**
sp1n	39	BFM-07	SIM-1	**Sch1**
sp2n	40	BFM-06	SIM-1	**Sch1**
sp3n	40	BFM-06	SIM-3	**Sch1**
sp4n	41	BFM-05	SIM-3	**Sch1**
sp5n	41	BFM-05	SIM-1	**Sch1**
sp6n	42	BFM-04	SIM-1	**Sch1**
sp7n	42	BFM-04	SIM-3	**Sch1**
sp8n	42	BFM-04	SIM-1	**Sch1**
sp9n	42	BFM-04	SIM-3	**Sch1**
sp10n	43	BFM-03	SIM-1	**Sch1**
sp11n	43	BFM-03	SIM-3	**Sch2**
sp12n	43	BFM-03	SIM-1	**Sch2**
sp13n	43	BFM-03	SIM-1	**Sch2**
sp14n	43	BFM-03	SIM-1	**Sch2**
sp15n	43	BFM-03	SIM-3	**Sch2**
sp16n	44	BFM-02	SIM-1	**Sch2**
sp17n	44	BFM-02	SIM-3	**Sch2**
sp18n	44	BFM-02	SIM-1	**Sch2**
sp19n	44	BFM-02	SIM-1	
sp20n	44	BFM-02	SIM-1	
sp21n	44	BFM-02	SIM-1	
sp22n	44	BFM-02	SIM-1	
scheduled				18

ILP Results			
mission remaining			sch?
sp1n	27	ACM-7	sch1
sp2n	28	ACM-6	sch1
sp3n	41	BFM-5	sch1
sp4n	42	BFM-4	sch1
sp5n	36	BFM-10	sch1
sp6n	30	ACM-4	sch1
sp7n	30	ACM-4	sch1
sp8n	35	BFM-11	sch1
sp9n	30	ACM-4	sch1
sp10n	41	BFM-5	sch1
sp11n	27	ACM-7	sch2
sp12n	42	BFM-4	sch2
sp13n	31	ACM-3	sch2
sp14n	31	ACM-3	sch2
sp15n	31	ACM-3	sch2
sp16n	32	ACM-2	sch2
sp17n	40	BFM-6	sch2
sp18n	27	ACM-7	sch2
sp19n	27	ACM-7	sch2
sp20n	32	ACM-2	
sp21n	32	ACM-2	
sp22n	44	BFM-2	
scheduled			19

The FTSSST Results				
MyOrder				**Sch?**
sp1n	27	ACM-7	SIM-1	**Sch1**
sp2n	28	ACM-6	SIM-1	**Sch1**
sp3n	41	BFM-05	SIM-3	**Sch1**
sp4n	42	BFM-4	SIM-3	**Sch1**
sp5n	36	BFM-10	SIM-1	**Sch1**
sp6n	30	ACM-4	SIM-1	**Sch1**
sp7n	30	ACM-4	SIM-3	**Sch1**
sp8n	35	BFM-11	SIM-1	**Sch1**
sp9n	30	ACM-4	SIM-3	**Sch1**
sp10n	41	BFM-05	SIM-1	**Sch1**
sp11n	27	ACM-7	SIM-3	**Sch1**
sp12n	42	BFM-04	SIM-1	**Sch2**
sp13n	31	ACM-3	SIM-1	**Sch2**
sp14n	31	ACM-3	SIM-1	**Sch2**
sp15n	31	ACM-3	SIM-3	**Sch2**
sp16n	32	ACM-2	SIM-1	**Sch2**
sp17n	40	BFM-06	SIM-3	**Sch2**
sp18n	27	ACM-7	SIM-1	**Sch2**
sp19n	27	ACM-7	SIM-1	**Sch2**
sp20n	32	ACM-2	SIM-1	
sp21n	32	ACM-2	SIM-1	
sp22n	44	BFM-02	SIM-1	
scheduled				19

Table 20. 9th and 10th Instances and Results

ILP Results

mission	remaining		sch?
sp1n	33	ACM-1	sch1
sp2n	33	ACM-1	sch1
sp3n	33	ACM-1	sch1
sp4n	33	ACM-1	sch1
sp5n	33	ACM-1	sch1
sp6n	33	ACM-1	sch1
sp7n	33	ACM-1	sch1
sp8n	33	ACM-1	sch1
sp9n	33	ACM-1	sch2
sp10n	33	ACM-1	sch2
sp11n	33	ACM-1	sch2
sp12n	33	ACM-1	sch2
sp13n	33	ACM-1	sch2
sp14n	33	ACM-1	sch2
sp15n	33	ACM-1	sch2
sp16n	33	ACM-1	sch2
sp17n	33	ACM-1	
sp18n	33	ACM-1	
sp19n	33	ACM-1	
sp20n	33	ACM-1	
sp21n	33	ACM-1	
sp22n	33	ACM-1	
scheduled			16

The FTSSST Results

MyOrder				**Sch?**
sp1n	33	ACM-1	SIM-1	**Sch1**
sp2n	33	ACM-1	SIM-1	**Sch1**
sp3n	33	ACM-1	SIM-3	**Sch1**
sp4n	33	ACM-1	SIM-3	**Sch1**
sp5n	33	ACM-1	SIM-1	**Sch1**
sp6n	33	ACM-1	SIM-1	**Sch1**
sp7n	33	ACM-1	SIM-3	**Sch1**
sp8n	33	ACM-1	SIM-1	**Sch1**
sp9n	33	ACM-1	SIM-3	**Sch2**
sp10n	33	ACM-1	SIM-1	**Sch2**
sp11n	33	ACM-1	SIM-3	**Sch2**
sp12n	33	ACM-1	SIM-1	**Sch2**
sp13n	33	ACM-1	SIM-1	**Sch2**
sp14n	33	ACM-1	SIM-1	**Sch2**
sp15n	33	ACM-1	SIM-3	**Sch2**
sp16n	33	ACM-1	SIM-1	**Sch2**
sp17n	33	ACM-1	SIM-3	
sp18n	33	ACM-1	SIM-1	
sp19n	33	ACM-1	SIM-1	
sp20n	33	ACM-1	SIM-1	
sp21n	33	ACM-1	SIM-1	
sp22n	33	ACM-1	SIM-1	
scheduled				16

ILP Results

mission	remaining		sch?
sp1n	27	ACM-7	Sch1
sp2n	28	ACM-6	Sch1
sp3n	28	ACM-6	Sch1
sp4n	29	ACM-5	Sch1
sp5n	29	ACM-5	Sch1
sp6n	30	ACM-4	Sch1
sp7n	30	ACM-4	Sch1
sp8n	30	ACM-4	Sch1
sp9n	30	ACM-4	Sch1
sp10n	31	ACM-3	Sch1
sp11n	31	ACM-3	Sch2
sp12n	31	ACM-3	Sch2
sp13n	31	ACM-3	Sch2
sp14n	31	ACM-3	Sch2
sp15n	31	ACM-3	Sch2
sp16n	32	ACM-2	Sch2
sp17n	32	ACM-2	Sch2
sp18n	32	ACM-2	Sch2
sp19n	32	ACM-2	Sch2
sp20n	32	ACM-2	
sp21n	32	ACM-2	
sp22n	32	ACM-2	
scheduled			19

The FTSSST Results

MyOrder				**Sch?**
sp1n	27	ACM-7	SIM-1	**Sch1**
sp2n	28	ACM-6	SIM-1	**Sch1**
sp3n	28	ACM-6	SIM-3	**Sch1**
sp4n	29	ACM-5	SIM-3	**Sch1**
sp5n	29	ACM-5	SIM-1	**Sch1**
sp6n	30	ACM-4	SIM-1	**Sch1**
sp7n	30	ACM-4	SIM-3	**Sch1**
sp8n	30	ACM-4	SIM-1	**Sch1**
sp9n	30	ACM-4	SIM-3	**Sch1**
sp10n	31	ACM-3	SIM-1	**Sch1**
sp11n	31	ACM-3	SIM-3	**Sch2**
sp12n	31	ACM-3	SIM-1	**Sch2**
sp13n	31	ACM-3	SIM-1	**Sch2**
sp14n	31	ACM-3	SIM-1	**Sch2**
sp15n	31	ACM-3	SIM-3	**Sch2**
sp16n	32	ACM-2	SIM-1	**Sch2**
sp17n	32	ACM-2	SIM-3	**Sch2**
sp18n	32	ACM-2	SIM-1	**Sch2**
sp19n	32	ACM-2	SIM-1	
sp20n	32	ACM-2	SIM-1	
sp21n	32	ACM-2	SIM-1	
sp22n	32	ACM-2	SIM-1	
scheduled				18

Bibliography

1. Albright, Christian S. *VBA for Modelers*, Pacific Grove, Wadsworth Group, 2001.

2. Baker, Kenneth R. *Introduction to Sequencing and Scheduling*, New York, John Wiley and Sons, 1974

3. Bottcher, J., Drexl, A., Kolisch, R. and Salewski, F. "Project Scheduling Under Partially Renewable Resource Constraints," Management Science, 1999, Vol. 45, 4: 543-559.

4. Bora, Ender NCO. Telephone interview, 143rd Oncel Squadron, Akinci AFB Ankara

5. Calhoun, Kevin M. " A Tabu Search For Scheduling And Rescheduling Combat Aircraft", Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, 2000

6. Clark, Wallace, *The Gannt Chart*, The Ronald press, 1952

7. Cummings, Steve. *VBA for Dummies,* New York Hungry Minds, 2001

8. Eddy, C. and Buchanan, T. *Microsoft Access 2000*, Sams Publishing, 1999

9. Eldem, Mehmet Col. Telephone interview, 143rd Oncel Squadron, Akinci AFB Ankara.

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

11. Feo, Thomas and Bard, Jonathan F. " Flight Scheduling and Maintenance Base Planning," Management Science, 1989, Vol. 35, 12: 1415-1432.

12. Gershkoff, Ira. "Optimizing Flight Crew Schedules," Interfaces, 1989, Vol19,4: 29-43.

13. Graves, G., McBride, R., Gershkoff, I. Anderson, D., Mahidhara, D. "Flight Crew Scheduling," Management Science, 1993, Vol. 39,6: 736,745.

14. Hooker, J.N. "Testing Heuristics: We Have It All Wrong," Journal of Heuristics, 1995, Vol.1: 33-42.

15. Jansen, Klaus and Porkolab, Lorant. "Improved Approximation Schemes For Scheduling Unrelated Parallel Machines," Mathematics of Operations Research, 2001, Vol. 26, 2:324:338.

16. Kivrak, Ozkan Capt. Telephone interview, 143rd Oncel Squadron, Akinci AFB Ankara.

17. Management Advisory Services Committee on Technical Studies of the American Institute of Certified Public Accountants, *Production Scheduling*, New York, American Institute of Certified Public Accountants, 1973.

18. "143rd Oncel Flight Training Squadron B Course Syllabus." January 2003

19. M'Hallah, R., Bulfin, R.L." Minimizing the Weighted Number of Tardy Jobs on A Single Machine," Elsevier, 2002, Article in Press

20. Microsoft Excel 2000, *On-Line Help,* 2000

21. Microsoft VBA 2000, *On-Line Help,* 2000

22. Muller-Merbach, Heiner. " Heuristics and Their Design: A Survey," EJOR, 1981,Vol.8: 1-23.

23. Nguyen, Cuong T. "An Interactive Decision Support System For Scheduling Fighter Pilot Training", Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, 2002

24. Pinedo, Michael. *Scheduling theory, Algorithms, and Systems*, New Jersey, Prentice-Hall, 2002.

25. Reeves, Colin R. *Modern Heuristic Techniques for Combinatorial Problems*, UK, McGraw Hill,1995

26. Shtub, A., J. Bard, S. Globerson. *Project Management Engineering*, Technology and Implementation, New Jersey: Prentice Hall, 1994

27. Silver, Edward A., Vidal, Victor V., and Werra, Dominique de. "A Tutorial On Heuristic Methods," EJOR, 1980, Vol. 5:153-162.

28. Stojkovic, Mirela and Soumis, Francois. "An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem," Management Science, 2001, Vol.47, 9:1290-1305.

29. Turkish Air Force "Standard Operating Procedures"

30. Turkish Air Educational Training Command "Flight Training Syllabus" January 2003

31. Van Hove, John C. *An Integer Program Decomposition Approach to Combat Planning*, Dissertation, Air Force Institute of Technology, Wright Patterson Air Force Base OH, 1998.

32. Varela, R., Vela, C., Puente, J. and Gomez, Alberto. " A Knowledge-based Evolutionary Strategy for Scheduling Problems with Bottlenecks," EJOR, 2001, Article in Press

33. Walkenbach, John. *Excel 2002 Power Programming with VBA*, New York, Hungry Minds, 2001.

34. Zanakis, Stelios H. and Evans, James R. "Heuristic "Optimization": Why, When, and How to Use It," Interfaces, 1981, Vol.11, 5: 84-90.

Vita

Captain Davut Aslan graduated from Kuleli Military High School in Istanbul, Turkey. He entered undergraduate studies at the Turkish Air Force Academy, Istanbul where he graduated with a Bachelor of Science degree in Electronics in August 1993.

His first assignment was at Sheppard AFB as a student in Undergraduate Pilot Training in January 1994. In Dec 1995, he was assigned to the 143rd F-16 Fighter training squadron, Ankara. In August 1996, he was assigned to 181st squadron and flew there for three years as a fighter pilot. Then he was reassigned to 143rd squadron in May1999 where he served as a support pilot for the flights. In Aug 2001, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the 143rd Fighter training Squadron.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 26-03-2003	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* Sep 2002 - Mar 2003

4. TITLE AND SUBTITLE A DECISION SUPPORT SYSTEM FOR EFFECTIVE SCHEDULING IN AN F-16 PILOT TRAINING SQUADRON	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Aslan, Davut, Captain, TUAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GOR/ENS/03-01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) 143rd Oncel Squadron 4th Main Jet Base Akinci-ANKARA	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Scheduling of flights for a flight training squadron involves the coordination of time and resources in a dynamic environment. The generation of a daily flight schedule (DFS) requires the proper coordination of resources within established time windows. This research provides a decision support tool to assist in the generation of the DFS. Three different priority rules are investigated for determining an initial ordering of flights and a shifting bottleneck heuristic is used to establish a candidate DFS. A user interface allows a scheduler to interact with the decision support tool during the DFS generation process. Furthermore, the decision support tool provides the capability to produce a weekly schedule for short-term planning purposes as well as the entire flight training program schedule for long-term planning purposes.
.

15. SUBJECT TERMS
Flight Scheduling, Daily Flight Schedule, Weekly Flight Schedule, Simulator Schedule, Multi- Modal Generalized Resource Constrained Project Scheduling, VBA Programming

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Victor, D. Wiley, Maj, USAF (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4284 ; e-mail: victor.wiley@afit.edu
U	U	U	UU	107	