United States Air Force Fighter Jet Maintenance Models: Effectiveness of Index Policies

by

John M Kessler

B.S., Mathematics and Operations Research (2011)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author ....
Sloan School of Management
May 17, 2013

Certified by ....
Retsef Levi
J. Spencer Standish Professor of Management
Associate Professor of Operations Management
Sloan School of Management
Thesis Supervisor

Certified by ....
Thomas Magnanti
Insitute Professor,Office of the Institute Professors
Thesis Supervisor

Accepted by ....
Patrick Jaillet
Dugald C. Jackson Professor, Department of Electrical Engineering and Computer Science
Co-director, Operations Research Center

The views expressed in this article 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.

United States Air Force Fighter Jet Maintenance Models: Effectiveness of Index Policies

by

John M Kessler

Submitted to the Sloan School of Management
on May 17, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

As some of the most technically complex systems in the world, United States fighter aircraft require a complex logistics system to sustain their reliable operation and ensure that the day-to-day Air Force missions can be satisfied. While there has been a lot of attention among academics and practitioners regarding the study of this complex logistics system, most of the focus has been on availability of spare parts that are indeed essential for the smooth operations of the fighter aircraft. However, in recent years there has been an increasing awareness that maintenance resources are an equally important enabler and should be considered together with inventory issues. The maintenance resources required to repair the fighter aircraft are expensive and therefore limited. Moreover, there are various types of maintenance that compete for the same resources. It is therefore imperative that the allocation of maintenance resources is done as efficiently as possible. In this thesis, we study two areas of fighter aircraft maintenance that could significantly benefit from improved resource allocation and scheduling strategies. We use quantitative and qualitative data from Air Force data-bases and logistics personnel to develop an innovative modeling framework to capture these challenging maintenance problems. This modeling framework is based on a generalization of the of the well-known multi-armed bandit superprocess problem. Using these models, we develop index policies which provide intuitive, easily implemented, and effective rules for scheduling maintenance activities and allocating maintenance resources. These policies seem to improve on existing best practices within the Air Force, and perform well in extensive data-driven simulated computational experiments. The first area is focused on the challenges of scheduling maintenance for the low observable (stealth) capabilities of the F-22 Raptor, specifically, maintenance of the outer coating of the aircraft that is essential to maintain its radar invisibility. In particular, we generate index policies that efficiently schedule which aircraft should enter low observable maintenance, how long they should be worked on, and which aircraft should fly in order to maximize the stealth capability of the fleet. Secondly, we model the maintenance process of the F100-229 engine, which is the primary propulsion method used in the F-16C/D and F-15E aircraft. In particular, we generate index policies to decide which engines should take priority over others, and whether or not certain components of the engines should be repaired or replaced. The policies address both elective (planned) and unplanned maintenance tasks.

Thesis Supervisor: Retsef Levi
Title: J. Spencer Standish Professor of Management
Associate Professor of Operations Management

Sloan School of Management

Thesis Supervisor: Thomas Magnanti
Title: Insitute Professor,Office of the Institute Professors

Acknowledgments

This thesis comes as a result of generous support and help from many people. I would first like to thank Professors Retsef Levi and Tom Magnanti for their patience and guidance over the past two years. I'm truly grateful for all that you have taught me, and for all the hours you've spent advising me. In addition, I'd like to thank Yaron Shaposhnik for all his help as my research colleague. I'll miss our Dunkin Donuts research discussions. I'd also like to thank Laura Rose and Andrew Carvalho for all their help over the past two years.

Next, I would like to thank the Air Force personnel who were so helpful in facilitating my questions and visits. In particular, I would like to thank 1st Lieutenant Alex Shamp and Mr. Greg Stonelake for their invaluable help in accommodating me during the work concerning LO maintenance. They not only provided useful data and insight, but facilitated visits to LO maintenance shops as well as entertaining hours of questions and conversation about the overall LO process.

I would also like to thank Mr. Brian Packard, Mr. Forunato Nepomuceno, and especially Senior Master Sergeant Charles Pierce for all their help teaching me about the engine maintenance process and providing data. They all took several hours (or even days!) answering my ceaseless questions for no reason other than their willingness to help me in my research. I truly appreciate it.

Next, I would like to thank Major Eric Zarybnisky for all his support. You essentially acted as my third advisor and spent countless hours on the phone guiding me. I would also like to thank Captain Eric Gons for his help in introducing me to many of the military members listed above, and for helping me to understand how to create effective analysis for the Air Force. I would also like to thank Professor Jack Muckstadt for his help and input in the engine maintenance models.

The research of presented in this thesis was supported by AFOSR awards FA9550-11-1-0150 and FA9550-08-1-0369.

Finally I'd like to thank all my friends in the ORC and my family. You all made the last two years some of the best yet.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The United States Air Force's ability to maintain a reliable, working fleet of aircraft is of paramount importance to its operational readiness. As an organization whose purpose is largely defined through air operations, it is essential that the Air Force maintains a fleet that is prepared to carry out any mission at a moment's notice. To keep Air Force pilots trained and qualified for their missions, flying operations occur regularly, even if no wartime or emergency situation require them. This high intensity of aircraft use, combined with the extreme technical and operational complexity of aircraft systems in general, requires maintenance operations that are efficient, effective, and able to handle any necessities that arise on a day to day basis.

Aircraft maintenance and the corresponding resource allocation are exceptionally complicated and difficult tasks, often requiring the cooperation of many organizations, extensive planning, and large amounts of equipment and parts. This is especially true of fighter aircraft. In addition, fighter aircraft are utilized in a manner that is conducive to mechanical failures and high levels of wear and tear. Whereas most larger jets tend to spend most of their flight time cruising at a constant speed and altitude, fighter jets are constantly altering both speed and altitude, and incur huge forces in the process. As some of the most technologically advanced systems in the world, fighter aircraft maintenance requires highly trained personnel, expensive equipment, and significant amounts of time to perform.

Due to the time consuming and expensive nature of fighter aircraft maintenance operations, it is imperative that maintenance resources are efficiently and appropriately deployed, allocated, and scheduled. Efficient scheduling can drastically increase the efficacy of a fixed amount of maintenance resources. Currently, much of the Air Force fighter aircraft maintenance scheduling and resource allocation is done in a very "ad-hoc" and inconsistent manner. The fact that there are so many variable factors to consider makes it very difficult to create a coherent and consistent policy that is always both feasible and efficient. Operational requirements of flying squadrons, equipment and parts inventory and availability, as well as competing types of maintenance all play a role in deciding when aircraft can be scheduled, and for what type of maintenance. In addition, most bases have different types of aircraft that use many of the same resources. Therefore, maintenance operations must be scheduled in a way that allows the different systems to share resources in the most effective way.

In this thesis, we will study two practically important types of maintenance. The first type of maintenance is called *low observable* (LO) maintenance and deals with the upkeep of stealth capabilities, and the second is called *modular engine maintenance*, which deals with the maintenance of specially designed engines that are comprised of a handful of modules.

The *low observable* maintenance is studied in Chapter 2 of this thesis. The newest generations of fighters have many characteristics that contribute to low observable, or stealth, capabilities. In particular, the F-22 radar invisibility is enabled by structural designs, but relies primarily on a smooth coating of specialized metallic paint, designed to minimize the radar signature of the aircraft to achieve these capabilities. This metallic paint degrades stochastically during flying operations, and a specialized type of maintenance known as low observable maintenance is required to repair the damages. However, in practice there is a relatively limited amount of low observable maintenance resources available. In addition, LO maintenance schedulers have many maintenance packages of varying length from which to choose when deciding which aircraft should be maintained. The LO state of the fleet is measured by a metric known as the *Fully Mission Capable* (FMC) rate, which is defined as the percentage of aircraft that have sufficiently low radar signatures so as to remain undetectable. The policies that are proposed in this thesis aim to help schedulers make daily decisions in order to keep the long term FMC rate as high as possible. In particular, we aim to provide rules concerning which aircraft should be maintained, for how long, and which aircraft should fly on a given day in order to sustain a high FMC rate.

The second problem, studied in Chapter 3, involves modular fighter engine maintenance scheduling. Many new fighter engines are built in a modular fashion, that includes a handful of major modules that can be removed and replaced with new modules if necessary. Each of these modules have a specified lifetime and, once expired, the module must be replaced; this is considered as an elective (predictable maintenance). In addition, modules can have unexpected breaks, which require maintainers to either repair the broken module, or replace it with a new module. Although it is often faster and easier to simply replace a module if there is one on hand, doing so is costly and will use a module that might have been better suited to replace a module that is almost out of life. Each operational base has an Air Force prescribed number of serviceable engines that must be in stock at all times, known as war ready engines (WRE). Whether or not a maintenance shop is able to fulfill its WRE requirement is one of the most important metrics upon which it is evaluated.

We aim to develop methods to help schedulers decide which engines to work on first, as well as decide which maintenance actions should be performed in order to minimize the average number of broken engines and maximize the proportion of time the WRE requirement is met. Specifically, we study these decisions in the context of the F100-229 engine which is a modular engine used primarily in the F-15E and F-16C/D fighter aircraft.

In each of the respective Chapters 2 and 3 we present a detailed description of maintenance operations and the associated decisions. These descriptions are based on relatively detailed, operational data obtained from Air Force maintenance units and databases, which is supplemented with qualitative information from meetings and discussions with active duty maintainers and program managers.

Both of the respective maintenance problems can be naturally modeled through dynamic programming (DP) formulations that capture the complex nature of the dynamic decisions under constraints over time. However, like many DP formulations, the respective state spaces would be very large and computationally intractable to manage. Moreover, the policies resulting from DP solutions often lack clear operational intuition.

Instead we show how these maintenance processes and decisions can be tractably modeled as variants of the *restless multi-armed bandit superprocess* [11]. The restless multi-armed bandit superprocess is a variant of the classic multi-armed bandit problem, which deals with the optimal allocation of resources amongst a number of competing projects. In a classic MAB problem, there are n projects and the state of each project in each period is known and belongs to a finite state space. In each time period, a decision must be made to "operate" one of the n projects. If a project is operated, then an immediate project and state specific reward is gained. Furthermore, the operated project transitions from its current state to a new state, according to a Markov rule that is project and state dependent. The unoperated projects in a given time period do not yield a reward and do not transition. The objective is to maximize the expected discounted reward over an infinite planning horizon. In the *multi-armed bandit superprocess* model, each bandit has many different modes of operation from which to choose. Finally, in the *restless multi-armed bandit superprocess*, we allow m of the n projects to be operated in each period. In addition, in the restless version of the MAB problem reward can gained from all bandits, and all bandits can experience a state transition, regardless of whether or not they are operated. We provide a detailed literature review and full description of the MAB problem, as well as some variants of the problem in Chapter 2.

When solved, multi-armed bandit models generate a decision making tool known as an *index policy*. Given a collection of projects and their states, an index policy assigns a numeric value measure to each possible action for each project, which gives decision makers an easy way to weigh the projects and actions against one another. In both Chapters 2 and 3, we adapt the LP-based relaxation technique first proposed by Bertsimas and Niño-Mora [6] to solve the MAB style models that we develop, and generate index policies that are effective, can be easily implemented, and provide valuable operational intuition.

Specifically, in the LO maintenance model, we devise two index policies; one which ranks the aircraft in terms of maintenance value, and one that ranks the aircraft in terms of flight value. These index policies can be used to easily determine which aircraft should be maintained, and which should be flown. In addition, the maintenance index policy indicates which of the many maintenance packages should be selected for each aircraft entering into maintenance. The policies are tested in an extensive data-driven simulation and are compared to existing policies and other more naive heuristics. The results demonstrate the effectiveness of the proposed index policies under realistic scenarios which are based on real data. In particular, the results indicate that using the proposed index policies instead of the currently implemented policies could raise the fleet FMC rate by as much as 20 percentage points, which would be a major operational improvement. Moreover, they provide significant organizational insight about the importance of coordinating flying and maintenance decisions.

In the engine maintenance model presented in Chapter 3, we propose a generalization of the

MAB model that includes multiple resources (as opposed to the traditional single resource), which are consumed in different combinations and rates depending on the actions taken. When solved, the model generates indices for each action-state pair, which can be thought of as "the value gained by performing action a on an aircraft in state s". However, since there now exist multiple resources, we cannot simply choose the m highest ranked aircraft and enter them into maintenance. For this reason, we propose two index heuristics which use the indices generated from the proposed model to make maintenance decisions. The first index heuristic uses the sign of each index to determine whether or not an action should be performed. The second heuristic disregards sign, but instead only considers broken engines for maintenance.

We then present data-driven simulation results that test the performance of the index heuristics against more naive policies. The results demonstrate the relative effectiveness of the proposed index heuristics under reasonably realistic scenarios. Specifically, when we compare the performance of a naive policy against the performance of the index heuristics we observe a decrease in the average number of unserviceable engines by almost 20%, a large decrease a decrease in variability, and a 98% decrease in the proportion of days where WRE requirements are unmet.

Our contributions in this thesis are threefold. First, we model important real world maintenance problems using a generalization of multi-armed bandit type models. Specifically, we present innovative frameworks for modeling the LO scheduling problem and the engine maintenance scheduling problem, which we believe can be extended to many other maintenance areas. In particular, we use generalizations of a restless multi-armed bandit superprocess model where the different actions can vary in length and are non-preemptive. In the engine maintenance chapter, we also introduce multiple different resources that can be consumed in different combinations depending on the action taken. We are unaware of any other work that utilizes models of this nature to address practical maintenance problems. Secondly, we show that the index policies generated by the proposed models can be effective in making maintenance decisions. In particular, we show that the proposed index policies can greatly outperform current policies. Finally, the proposed index policies are intuitive and provide lucid insight into both areas of maintenance operations.

1.1 Literature Review

In this section we provide a brief summary of the existing literature related to the aircraft maintenance problem. We begin by considering published Air Force policies and doctrine concerning maintenance scheduling. We will then describe the literature concerning scheduling of modular engine maintenance scheduling and low observable maintenance scheduling. Finally, we will review some of the existing multi-armed bandit literature.

1.1.1 High Level Maintenance Policy

High level Air Force maintenance policies can be found in Air Force Instruction (AFI) 21-101, "Aircraft and Equipment Maintenance Management" [1]. This document outlines high level prioritization of maintenance actions, what to include in schedules, and other guidelines for scheduling in a military aviation framework. AFI 21-101 is not aircraft specific, but rather provides general guidelines on how flying squadrons and maintenance squadrons interact, the different positions and responsibilities associated with these squadrons, and inventory management doctrine. Much of the aircraft specific guidance is contained in Technical Order (TO) documents and are not publicly available. AFI 21-101 stresses the idea of cyclical maintenance and the importance of annual, quarterly, monthly, and weekly schedules. Annual and quarterly requirements are provided to maintenance supervision, and a rough schedule is designed to meet these requirements. These schedules are more of a plan on how much work the maintenance group should expect in the next quarter/ year, rather than a traditional schedule. Monthly and weekly schedules are revisions of the year and quarter schedules, and deal more with the specifics (take off and landing times, armament requirements, etc.).

Air Combat Command Instruction (ACCI) document 21-165 contains guidelines aimed specifically at the fighter aircraft community [2]. This document outlines high level guidelines for how to schedule fighter aircraft flight and maintenance operations. It reiterates the importance of the different planning cycles (annual, quarterly, monthly, and weekly) and what is expected for each of these cycles. ACCI 21-165 stresses the importance of "planning what you fly, and flying what you plan", and working together with all the groups that are part of successfully completing flying missions. However, no specific guidance is given as to which aircraft should be chosen when, or what type of maintenance to perform. Often, these decisions are left to the discretion of the schedulers, who base their decisions on years of experience in the field.

1.1.2 Low Observable Maintenance

There is relatively very little existing work related to F-22 low observable (LO) maintenance. This is likely because the technology is still relatively new and unique to the F-22 (other aircraft have stealth capability, but the way they achieve it is different). The few documents that address LO maintenance address the problem in terms of how to perform the maintenance, as opposed to how the maintenance resources should be scheduled and utilized. For example, Ysebaert et al. [30] discuss methods of decreasing maintenance workload by opting to replace panels and working on damaged panels offline. However, they do not address the choice of aircraft to put into maintenance at a given time, nor do they discuss what type of work should be conducted once an aircraft has been selected.

1.1.3 Engine Maintenance Scheduling

Engine maintenance related problems have been studied in much greater depth than the LO problem, largely due to the fact that efficient aircraft maintenance scheduling is of great interest in the commercial airline industry as well as in the military. However, many of the models or plans that are developed for commercial use are often not applicable to the military realm. One of the most significant reasons for this has to do with the different ways flight operations are performed in each organization. In a commercial organization, an airline owns aircraft and wants to route them on flights and have them maintained wherever it is most efficient. Air Force aircraft belong to a specific base and ideally have most of their maintenance performed at that base. This is especially true of fighter aircraft, where most of the flying operations performed begin and end at the same base. Many papers contain models for the commercial airline maintenance [23, 4, 14], but none of these overlap very much with the modular fighter engine maintenance problem.

Much of the research performed specifically in the area fighter engine maintenance is concerned with efficient inventory policies. For example, Muckstadt[20] presents a model that attempts to determine efficient stock levels of engine components across a network of locations in a multi-echelon system. In another work, Jackson et al. [16] create a model that helps determine which locations in a repair network should be resupplied, as well as the transportation method that should be used to perform the replenishment of inventory at each location being resupplied. Zarybnisky[31] also contributes to this area by developing a model that balances the tradeoffs between inventory, maintenance capacity, and transportation resources. A more comprehensive account of the literature concerning inventory policy for Air Force fighter engines, as well as other modular-style systems can be found in "Analysis and Algorithms for Service Parts Supply Chains" by J. Muckstadt[21].

There is also some work related to fighter engine maintenance scheduling, though it is less common than its commercial counterpart. Kleeman et al. [17] provide an interesting analysis of scheduled maintenance and component matching for fighter engines. This, however, does not deal with any unscheduled breakages. . Finally, Levi et al. [19] also study the problem of preventative maintenance scheduling. In particular they suggest the use of a *shifted power-of-two policy* which involves rounding module down to the nearest power of two and maintaining each module at intervals of this power of two. Although this policy is effective and generally performs within a few percentage points of optimality, this work does not suggest methods for the incorporation of unscheduled maintenance actions, which rely on the same resources as preventative maintenance actions. Therefore, the work presented in Chapter 3 regarding fighter engine maintenance is unique in incorporating scheduled maintenance, unscheduled maintenance, and the modular nature of many fighter engines

1.2 Organization and Outline

The remainder of the work presented in this thesis is structured as follows:

Chapter 2

In this chapter, we focus on relatively new maintenance and operational scheduling challenges that are faced by the United States Air Force concerning low-observable (LO), or stealth, aircraft. The LO capabilities of an aircraft degrade stochastically as it flies, making it difficult to make maintenance scheduling decisions. Maintainers can address these damages, but must decide which aircraft should be put into maintenance, and for how long. Using data obtained from an active duty Air Force F-22 wing and interviews with Air Force maintainers and program specialists, we model this problem as a generalization of the well-known restless multi-armed bandit superprocess. Specifically, we extend the traditional model to allow for actions that require varying lengths of time, and generate two separate index policies from a single model; one for maintenance actions and one for the flying action. These index policies allow maintenance schedulers to intuitively, quickly, and effectively rank a fleet of aircraft based on each aircraft's LO status and decide which aircraft should enter into LO maintenance and for how long, and which aircraft should be used to satisfy daily sortie requirements. Finally, we present extensive data-driven, detailed simulation results where we compare the performance of the index policies against policies used by the Air Force, as well as some other possible heuristics. The results indicate the strength of the index policies significantly outperform existing policies. In particular the experiments highlight the importance of good flight decisions.

Chapter 3

In this chapter we shift our focus to U.S. Air Force fighter aircraft engine maintenance scheduling. We present background concerning how fighter engine maintenance is performed, the options available to maintainers, and current practice. We then introduce a restless multi-armed bandit superprocess model with multiple resources to generate an index policy. This index policy allows us to rank aircraft in terms of what type of maintenance should be performed on which aircraft at a given time. Since our index policy now includes actions that consume multiple different resources, we propose two methods to make aircraft maintenance decisions using our index policies. Finally, we conduct an empirical analysis on our policies. We find that again our policies outperform naive policies by increasing the number of working engines on average and help to alleviate spikes in demand.

Chapter 4

Conclusion and future work

Chapter 2

Low Observable Maintenance

2.1 Introduction

In this chapter, we focus on relatively new maintenance and operational scheduling challenges that are faced by the United States Air Force concerning low-observable (LO) aircraft. The F-22 Raptor is one of the newest fighter jets in the Air Force inventory and has LO capabilities that make it invisible to radar. These capabilities are obtained largely through a special metallic coating on the exterior of the aircraft, which has unique maintenance requirements that did not exist for previous generations of fighter aircraft. In particular, this coating degrades stochastically, both in magnitude and physical location. The radar signature of an aircraft can be measured with a specialized computer system, that assigns a numeric evaluation to how much each specific damage contributes to overall radar signature. Using the information gained from this system, maintainers make decisions on which aircraft should enter the limited number of LO maintenance bays to be worked on, as well as how long these aircraft should be maintained for. At the same time, aircraft must be scheduled to meet daily sortie requirements.

The modeling work in this chapter is based on quantitative data from an operational F-22 unit concerning radar signature increases due to damages to the LO coating over a two and a half year time span. In particular, the data contains the radar signature increases for each aircraft that flew on a given day, as well as past LO maintenance schedules. These schedules provide data on the radar signature decreases that occur as a result of LO maintenance. In addition, we supplement the data with qualitative information from meetings and discussions with active duty LO maintainers and program managers concerning the overall LO maintenance process. This helped to develop a relatively accurate and realistic model for the problem as a generalization of the well known *restless multi-armed bandit* problem with multiple controls (sometimes called a *superprocess*). Specifically, we use an extension of the traditional restless model to allow for actions that require varying lengths of time, and generate two separate index policies from a single model; one for maintenance actions and one for the flying action. We accomplish this by using a variant of the LP relaxation of the MAB problem developed by Bertsimas and Niño-Mora [6]. The resulting index policies allow schedulers to quickly and easily rank a fleet of aircraft based on each aircraft's LO status and decide which aircraft should enter into LO maintenance, how many days they should be maintained, and which aircraft should be used for the daily sorties.

In Section 2.4 we provide detailed literature review on the various multi-armed bandit models and policies that were studied in the past. While the MAB has been used to study various decision making problems such as internet advertising [3] and task scheduling on cloud resources [15], there exist relatively few applications to maintenance. Index policies have been proposed as a scheduling method for maintenance operations by Glazebrook et al. [13] who proposed using index policies to make maintenance decisions in a scenario with M non-identical machines and R repair men. However this is only a theoretical model with no concrete real life application.

Finally, we present extensive data-driven simulation results in which we compare the performance of the index policies against policies currently used by the Air Force and other possible more naive heuristics. The results indicate that the model driven index policies outperform existing scheduling policies by a large margin. In particular, the simulation predicts that the use of index based policies rather than the currently implemented policies can cause fully mission capable (FMC) rates to jump by more than 20 percentage points (we note that FMC is perhaps the single most important performance metric by which operational Air Force units are evaluated). These results also show that coordinated maintenance and flight decisions have a large impact on FMC rates, and that efficiently selecting which aircraft are used to satisfy the daily sortie requirements can significantly improve system performance. In contrast, currently LO concerns are not generally considered when selecting which aircraft will fly on a given day.

The contributions of this chapter are threefold. First, we model an important real world maintenance problem using multi-armed bandit type models. Specifically, we present an innovative framework for modeling the LO scheduling problem that we believe can be extended to many other maintenance problems. In particular, we use a generalization of a restless multi-armed bandit superprocess model where the different actions can very in length and are non-preemptive. We are unaware of any other work that utilizes a model of this nature. Secondly, we show that the index policies generated by the proposed model can be effective in making LO maintenance decisions. In particular, we show that the proposed index policies can greatly outperform current policies. Finally, the proposed index policies are intuitive and provide lucid insight into which aircraft should be maintained, and which should be flown. Moreover, they provide significant organizational insight about the importance of coordinating flying and maintenance decisions.

In Section 2.2 we begin by presenting a detailed explanation of the LO degradation and maintenance processes and the dynamics associated with them. In Section 2.3 we describe the operational data we will use as part of our modeling process. Next, we present the multi-armed bandit problem, some generalizations of this problem, and present relevant literature in Section 2.4. We then show how to model the LO process and associated decisions as a generalized restless MAB superprocess problem in Section 2.5, and show how to generate index policies from this model in Section 2.6. In Section 2.7, we apply our index policies using simulation and compare their performance against approximations of current policies.

2.2 Low-Observable Maintenance Process

The newest generations of fighter aircraft in the Air Force, such as the F-22, have low-observable technologies that present new maintenance and operational challenges that did not exist for previous generations of fighter aircraft. In particular, the outer surfaces of the LO aircraft are coated with a metallic paint that is designed to minimize the radar signature of the aircraft. While LO aircraft have many design features that contribute to the LO capability of the aircraft, such as the shape and angles of the aircraft, the exterior coating is the primary contributor to the increased maintenance requirements for LO aircraft. If the coating is damaged in any way, the radar signature of the aircraft will likely be affected. Since LO aircraft are not considered to be fully mission capable (FMC) unless their radar signature is below a certain level, maintenance personnel must continuously repair the metallic coating on LO aircraft in order to sustain an acceptable FMC rate for a fleet of aircraft.

To properly plan LO maintenance, maintenance personnel must be able to characterize the state of each aircraft's LO capabilities. As an aircraft accumulates minor damages, the LO capability of the aircraft deteriorates. Depending on the size, location, and shape of each specific damage, the overall impact of a single damage can range from being negligible to causing the aircraft to no longer be FMC. Each day an aircraft flies, maintenance personnel record all new damages into the *Signature Assessment System* (SAS). To estimate the cumulative impact of all the damages, the SAS measures the LO capability of an aircraft through a metric called a *SAS number.*

The SAS number is crucial in making LO maintenance decisions. Using the damages information that is input into the database, a SAS number is calculated and quantifies the state of an aircraft's LO capability. A higher SAS score corresponds to a larger radar signature and a lower SAS number corresponds to a smaller radar signature. The minimum possible SAS number is 0 which represents an aircraft with maximum stealth capabilities. If an aircraft's SAS number exceeds 100, then the aircraft is no longer considered to be FMC since its stealth characteristics have been reduced. It is important to note however, that even if an aircraft is no longer FMC due to its high SAS number, it can still be flown in support of daily sortie requirements. This is due to the fact that the damages that increase the SAS number do not have a significant impact on the flight characteristics of the aircraft. Just as each aircraft is measured by its SAS number, the overall state of a fleet of aircraft is also measured in terms of the FMC rate. The LO portion of the FMC rate represents the percentage of aircraft that currently have a SAS number below the FMC threshold of 100. Therefore, maintenance and sortie schedulers must carefully monitor the SAS number for each aircraft and work to keep each aircraft's SAS number below the FMC threshold by performing preventative maintenance.

Damages to the LO coating primarily occur in two ways. The first pertains to the case when an internal component of an aircraft needs to be maintained and therefore requires that an outer panel be removed in order to access the inside of the aircraft. Since the coating on the external surfaces of the aircraft is continuous and smooth across all panels, LO maintenance personnel must break the metallic coating around the edge of the panel to remove the panel, in an action referred to as *picking*, and then must restore the coating when the panel is reinstalled, in an action known as *facilitate other maintenance* (FOM). Of these activities, picking is an efficient process while FOM requires much more time and effort to complete. It is important to note that when a maintainer performs

FOM maintenance actions, they are utilizing LO maintenance resources (personnel, maintenance bays, etc.) without fixing any of the current LO damage on the aircraft.

The second form of LO damage comes from the deterioration of the metallic coating due to flying activities. Simple scrapes and dings can have a significant impact on the overall radar signature of an aircraft depending on size and location. Therefore, LO maintenance personnel must carefully track the damages on each aircraft and decide when to repair them. Maintenance to address this type of damage is solely aimed at reducing the radar signature of an aircraft and improving its LO capabilities. We will refer to these maintenance actions as LO reduction, or *LO redux*, actions.

Generally, maintainers have two different methods of LO redux. The first method of redux maintenance is referred to as *long lane maintenance*. The long lane consists of an eleven day tear down of the aircraft, where the external surface of the aircraft is completely overhauled and refurbished. This type of maintenance, though very time intensive, fixes any LO damages addressed and eliminates most of the residuals.

The second method can be considered localized maintenance. Many of these fixes are performed by a maintainer with a paint brush and some metallic paint. However, any overlap or inconsistencies in the paint will increase radar signature. Because of this, many of the maintenance actions performed in normal redux leave a residual amount of radar signature, dubbed *residuals*. Residuals are typically small in terms of SAS impact, and can generally be fixed only using long lane maintenance.

When deciding which aircraft to enter into SAS redux, schedulers must consider not only the SAS number for each aircraft, but also the nature of the damages that make up the total SAS number. Although two aircraft might have the same SAS number, the location and distribution of damages on each aircraft could be drastically different. One aircraft might have only a few damages that each cause a significant increase in the SAS number while the second aircraft might have a large number of damages that each contribute a small amount to the overall SAS. Individual damages are classified as either a "heavy hitter" (HH) or a "non heavy hitter" (NHH), where a heavy hitter is a damage that single-handedly increases the SAS number of an aircraft by a significant amount.

Depending on the distribution of HH and NHH damages on an aircraft, the amount by which the aircraft's SAS number will decrease after a fixed time in SAS redux maintenance will vary drastically. This reduction in SAS number is referred to as the *buy back*. Maintenance personnel generally repair damages in their order of contribution to the overall SAS number, repairing the damages that contribute the most to the SAS number first. Furthermore, most damages take approximately equal amounts of time to repair regardless of how much each damage contributes to the overall SAS number. Therefore, if an aircraft has a large number of heavy hitters, a single day of SAS redux will result in a large buy back. If the same aircraft undergoes SAS redux for multiple days, each subsequent day will have decreasing marginal return in terms of SAS buy back.

It may then seem pertinent to only complete SAS redux on aircraft with heavy hitters since they offer the most buy back per day of SAS redux. Notice however, that if only heavy hitters are repaired, the FMC rate of the fleet may increase in the short term, but the number of smaller damages on each aircraft will slowly increase over time and will push the SAS number closer to the FMC threshold. All aircraft will then have to undergo extensive SAS redux in order to repair the damages that had long been ignored. On the other hand, if schedulers try to maintain every damage as soon as it occurs, the limited maintenance capacity will be backlogged and the FMC rate will likely suffer. Therefore, schedulers should carefully choose when to maintain each aircraft and to what extent, in order to sustain a high FMC rate. In general, schedulers can assign an aircraft to undergo a SAS redux action for durations that range from one to four days, or a long lane which takes eleven days.

Maintenance capacity and sortie requirements are the limiting factors in the LO maintenance process. Each day, a given number of sorties are required to occur which causes the SAS number of the aircraft that fly to stochastically increase. At the same time, there is a fixed capacity for LO actions due to facilities, equipment, and manpower. The same resources are used to perform SAS redux as well as FOM mentioned earlier. Thus, SAS redux scheduling must adapt to changing levels of available LO resources as current policy dictates that FOM must be completed before an aircraft can return to operations and thus takes precedence over SAS redux in most cases.

Schedulers are then faced with the difficult challenge of deciding which aircraft should fly and potentially increase their SAS number, and which aircraft should use the limited available LO redux resources and for how long. Currently, LO status almost never plays a part in scheduling which aircraft will fly sorties. Generally speaking, the flying squadron is responsible for selecting which aircraft will fly.

Although the LO maintenance shop has influence on which aircraft it receives for maintenance, decisions regarding SAS redux are largely dependent on the personalities of each LO maintenance unit. Since there is little published guidance regarding LO maintenance, each maintenance unit has the flexibility to make LO maintenance decisions however they see fit. As a result, the LO maintenance policies used can vary between locations. In speaking with several experienced maintenance personnel, the LO maintenance decision process was described as being "somewhat haphazard".

In addition, the action of taking an aircraft out of service to put into maintenance is not one that the LO maintenance shop can make independently. Occupying aircraft for the purpose of LO maintenance must be coordinated with the flying squadron as well as other maintenance shops. In general, most LO maintenance shops tend to use their limited influence to request aircraft with heavy hitters and take whatever aircraft might be available after that.

2.3 SAS Evolution Data

In this section, we present data pertaining to the SAS evolution as it pertains to LO redux. At the end of each day an aircraft is flown, an LO post flight inspection is completed. Maintenance personnel inspect the aircraft for new damages to the LO coating and record this information in the SAS database. The pre- and post-flight SAS numbers are compared to determine the SAS increase due to that day's flights.

We begin by analyzing the empirical probability distribution of the daily SAS increase based on over 5000 data points over 2.5 years of operations. Table 2.1 shows specific probabilities of SAS increases between 0 and 10. Notice that the vast majority of SAS increase are relatively small. Approximately half the days an aircraft flies, the SAS increases by 1 unit or less, and approximately 83% of the time the SAS increase is 10 units or less. Figure 2.3.1 shows the empirical probability mass function of the remainder of the SAS increase data. Of note in this graph are the SAS increases of 154-160. These increases can be attributed to a specific type of damage, specifically a *canopy damage*, that immediately increases the aircraft's SAS number by approximately 150 units. A canopy damage is anytime the aircraft canopy coating is broken in any way. Due to the location and shape of the aircraft canopy, any break in the LO coating on the canopy results in large SAS increases.

Although the data provides us with a very good understanding of how the total SAS number for each aircraft evolves over time, the data does not provide any information regarding the distribution

Table 2.1: SAS Increase Probabilities.

SAS Increase	Probability	Cumulative Prob.
0	0.326	0.326
1	0.194	0.520
2	0.107	0.627
3	0.056	0.683
4	0.034	0.717
5	0.039	0.755
6	0.021	0.776
7	0.018	0.794
8	0.013	0.807
9	0.013	0.820
10	0.009	0.829

Figure 2.3.1: Truncated PMF of daily SAS increases.

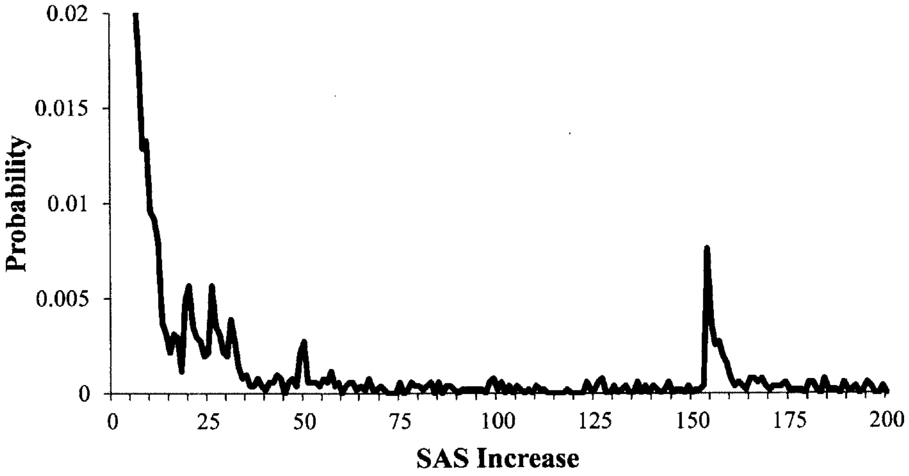


Table 2.2: SAS Increase Correlation Coefficients.

	AC Number	Number of Flights	Flight Hours	Start SAS
SAS Increase	-0.071	0.016	0.045	0.102

of damages on each aircraft. For example, the data may show that an aircraft had a preflight SAS number of 70 and a postflight SAS number of 90. If the SAS increase was all due to one damage, then we would know that there is a significant HH. If the 20 unit increase was due to several damages, then the aircraft may not have a HH. However, the data does not specify the number or distribution of damages that caused the SAS increase as this data is confidential. Based on the data, specifically the significant amount of mass for low SAS increases, we assume that if an aircraft experiences an increase of 20 units or more from a single day, it has at least one HH. Under that assumption, which LO maintainers verify as being fairly realistic, the probability of a HH on any given day of flying is approximately 13%.

To fully understand the stochastic evolution of an aircraft's SAS number, we examine which factors influence the SAS increases experienced by an aircraft. Table 2.2 shows the correlation coefficients between the daily SAS increase and a number of different variables. Intuition would potentially suggest that if an aircraft accrues more flights and flight hours in a day, the daily SAS increase would be higher. However, the correlation coefficients for these two variables, 0.016 and 0.045, respectively, are insignificant. Further analysis using regression models show that factors, such as aircraft tail number, number of flights, total flight hours, days since last redux action, and flights since last redux action are all insignificant. Since there are no apparent factors that influence the distribution of SAS increases that are directly related to SAS redux and flight scheduling, for the remainder of this chapter we assume that SAS increases are random and follow the empirical distribution shown in Table 2.1 and Figure 2.3.1.

Based on the lack of correlation between SAS increase and relevant factors, we assume that the probability distribution of the SAS increases is independent of the state of a given aircraft. That is, whether an aircraft has a SAS number of 20 and no heavy hitters, or a SAS number of 150 and multiple heavy hitters, the probability of the SAS number of either aircraft increasing by the same amount is equal.

While the upward transitions of the SAS number are relatively well characterized by the data, the downward transitions that result from SAS redux maintenance actions are not as well understood.

In reality, maintainers can use the signature assessment system to see which hits they should address on each aircraft and exactly how much SAS buyback they will gain by addressing these damages. However, our data is limited in this respect and only provides total SAS numbers for aircraft, not the distribution of damages. Since a single SAS number can represent an almost infinite number of damage distributions across an aircraft, it is difficult to determine exactly how much buyback we will achieve from a maintenance action, given only a SAS number. Later in Section 2.4 we discuss how to model this.

2.4 Multi-Armed Bandit Problem Summary and Relevant Literature Review

In the previous sections we detail the LO maintenance process and present relevant data. One possible way to model the decision problem described above is through a dynamic programming formulation. However, the resulting formulation is not likely to be tractable. In particular, to capture the maintenance capacity constraint within a DP, the state of the system would include information regarding every aircraft in the fleet. In addition, maintenance actions can last multiple periods which also increases the size of the state space. For each aircraft, information about the SAS number and distribution of damages would be necessary. Given that, in practice, there can be approximately 40 aircraft in a fleet of aircraft and SAS numbers can take on a wide range of values, the size of the state space necessary to capture the relevant information about the system would easily make the DP practically intractable. In addition, even if the DP could be solved optimally it is not likely to devise policies that will be operationally simple or realistic to implement in practice.

Due to the issues involved in modeling the LO maintenance scheduling problem as a dynamic program, we are motivated to find another method for guiding both LO maintenance and flight scheduling decisions. In this section, we describe the well-known multi-armed bandit (MAB) problem as a tractable modeling alternative to a DP.

A *multi-armed bandit problem* deals with the optimal allocation of resources amongst a number of competing projects [11]. In a classic MAB problem, there are n projects and the state $s_1, s_2, ..., s_n$ of each project $i = 1, 2, ..., n$ is known. In each time period, a decision must be made to "operate" one of the n projects. If project i is operated, then an immediate reward of $g_i(s_i)$ is gained. Furthermore, project i transitions from state $s_i(t)$ to $s_i(t+1)$ according to a Markov rule that is project and state dependent. The unoperated projects in a given time period do not yield a reward and do not transition. The objective is to maximize the expected discounted reward over an infinite planning horizon. Gittins [12] showed that an index based policy is optimal for the classic MAB problem. In particular, each possible state of a project is assigned an index value and in a given time period, the project with the highest index is operated. Whittle [29] uses a dynamic programming approach to prove the same index result as Gittins and uses this framework to extend the optimality of the index rule to a special case called the *multi-armed bandit superprocess.* The MAB superprocess is the same as the classical MAB except that each project now has a number of feasible controls, from which to choose. However, the traditional MAB superprocess problem still allows only one bandit to be chosen in each period. Varaiya et al. [25] present a variant of the MAB superprocess which includes non-preemptive actions that continue for longer than a single period, and show index policies to be optimal in this case.

The *restless multi-armed bandit* is a generalization of the classic MAB problem. Unlike the classic MAB where the unoperated projects remain static (i.e., did not undergo a state transition) and do not contribute to the objective function, in the restless case, both operated and unoperated projects undergo a state transition, in each time period, and both yield an immediate reward. In addition, in the restless MAB we no longer restrict ourselves to operating one project at a time. Rather, we allow for up to m of the n projects to be operated at any given time. There is a natural generalization of an index policy to this setting, in which the n projects with the highest states are chosen in each period. While the index policy of Gittins [12] is no longer guaranteed to be optimal, Whittle [28] generalizes the Gittins index for the restless case. Weber et al. [27] showed that this policy is asymptotically optimal, in the regime where m and n go to infinity while $\frac{m}{n}$ remains fixed. Finding an optimal control is unlikely for most cases, as Papadimitriou et. al [22] showed that even with deterministic transition rules the restless bandit problem is PSPACE-hard.

Whittle's heuristic leads to a simple index policy that is used to decide what projects to operate at any given time. The heuristic hinges on relaxing the constraint on the number of projects that can be operated at one time. Recall in the restless MAB, m of n projects could be operated in each period. By requiring that this constraint be met in expectation rather than in each period and then Lagrangifying the relaxed constraint, Whittle showed that the Lagrange multiplier could be used as an index value [28]. In general, the heuristic considers the single bandit problem for each possible state and seeks a subsidy λ_{s_i} that makes it equally attractive to operate or not operate a project i in state s_i. This can be done independently for each project. The projects are then ranked according to their current state and the respective subsidy values. In each period, the m highest ranked projects are operated. Notice that this leads to a simple index policy that is easily implemented and followed.

Although the Whittle heuristic is significantly more tractable than modeling the problem as a large DP, it still requires significant computation. Bertsimas and Niño-Mora [6] introduce a sequence of LP relaxations that approximate the restless MAB problem. These linear programs use primal-dual heuristics to determine which bandits should be worked in each time period. If there are no isolated states (i.e., from any starting state, it is possible to reach every other state in a finite number of periods), the primal-dual heuristic reduces to a priority-index rule. This rule computes an index based on the difference between the reduced costs of the operate and not operate variables for each bandit. The bandits with largest indices are then chosen for operation.

In contrast to the dynamic programming based index heuristic presented by Whittle, which performs a large offline calculation to produce index values for all possible states, the linear programming (LP) uses current information about the state of all aircraft. The LP is solved in each period for the states that are realized on the specific sample path. This approach is strongly based on the linear programming formulation of a dynamic program introduced by Bertsekas [5]. The first-order relaxation is based on the dynamic programming formulation developed by Whittle [28]. Higher order relaxations add additional variables and constraints to better approximate the true restless MAB process. While the first-order LP relaxation is polynomial in size, higher order relaxations, though tighter, quickly become intractable due to the number of variables and constraints.

The LP heuristic presented by Bertsimas and Niño-Mora [6] has been extended in more recent work. In particular, Glazebrook et al. [10] apply the LP heuristic to a MAB superprocess model. By using the reduced costs from the solved LP, Garbe and Glazebrook find the best action for each bandit/ state pair. They then extend the primal dual heuristic introduced by Bertsimas and Niño-Mora to decide which bandits should be worked on. Again, this primal-dual heuristic can be reduced to an index policy. However, they do not consider cases with multiple resources, or actions that can span multiple periods.

In Section 2.5 we will describe how to use the MAB superprocess modeling framework to model the LO scheduling problem. In Section 2.6, we will show how to adapt these approaches to compute effective policies for the LO scheduling problem. We note that index policies are attractive in our setting since they are simple to communicate and are operationally intuitive. In fact, we will show that the index policies that we compute can be described by intuitive rules.

2.5 Restless MAB Superprocess Modeling Framework

In this section, we show how to model the LO scheduling problem as a generalized restless MAB superprocess. Then in Section 2.6, we apply a variant of the LP based index heuristic of Bertsimas and Niño-Mora [6] to the LO maintenance scheduling problem. The LO maintenance scheduling problem matches very closely to the construct of the MAB superprocess problem discussed in the previous section. Each aircraft can be thought of as a project and deciding to maintain or fly an aircraft is analogous to operating a project. For aircraft that are not assigned to maintenance or flying, the state of the aircraft will remain unchanged. For aircraft that are maintained or flown, the transition probabilities are also known based on the state of the aircraft. The maintenance capacity constraint is analogous to being allowed to operate only m out of the n projects, in each period. Furthermore, the reward for operating an aircraft is a function of the aircraft's SAS number. By providing a reward to aircraft that are below the FMC threshold and then maximizing the discounted expected reward over an infinite horizon, we can maximize the sustainable FMC rate for a fleet of aircraft.

There are two primary differences between the LO maintenance scheduling problem and the typical MAB superprocess problem. The first is that SAS redux actions can last multiple periods. In the MAB context, this implies an action in one period can dictate resource consumption in subsequent periods. Second, we wish to generate two different index policies from our model; one for maintenance actions, and one for the flying action. Each of these sets of indices (maintenance and flying) relate to a different "resource". The maintenance actions are limited by the typical capacity constraint (this is a "packing" constraint), whereas we require a certain number of flying actions each period to satisfy the daily sortie requirement (this is a "covering" constraint).

2.5.1 State Space

Our objective in modeling the LO problem is to form a tractable state space that captures the relevant information about aircraft. Recall in Section 2.3, we stated that the upward transitions in the SAS number are captured by the empirical distribution from the historical data and factors related to LO redux scheduling do not influence the upward transition probabilities. For downward transitions in the SAS number as a result of SAS redux, the transition probabilities are dependent upon the SAS number of the aircraft when it entered maintenance as well as the distribution of damages on the aircraft. Therefore, the state space of the model must include information regarding the SAS number of each aircraft as well as some information regarding the severity of damages on each aircraft.

Along with the SAS number and distribution of damages, we must be able to capture the state of an aircraft that is in maintenance. There are multiple types of SAS redux actions with different durations, so the model must track how many days are remaining in maintenance.

Given these consideration we allow the state, s_i, of each aircraft i to be defined by four pieces of information:

- the total SAS number of the aircraft (0-300),
- whether or not the aircraft has a HH (0 or 1),
- total amount of residual SAS (0-100)
- if in maintenance, the number of days remaining in maintenance (0-11).

Therefore, s_i is a vector that contains four pieces of information. The first value will be an integer value in the range $[0, 300]$ and will represent the SAS number of the aircraft. Although the SAS number can actually take on values greater than 300, we limit it due to the fact that aircraft with SAS numbers exceeding 300 are rare. The second value of the state vector is a binary indicator that represents whether an aircraft has a heavy hitter or not. Although this is a highly generalized representation of the distribution of damages on an aircraft, the seemingly simple distinction between an aircraft with a HH and one without a HH allows us to much more accurately represent the state and the state transitions without greatly increasing the size of the state space. This also matches the intuition of many practitioners that believe the mere existence of a HH would be the most important fact in making maintenance decisions. If historical data was available regarding the specific sizes, locations, and SAS impact of each individual damage, the state space could be expanded further and the LO maintenance scheduling process could be more accurately modeled. However, this would potentially affect the tractability of the problem.

The third value represents the amount of SAS damage that is residual damage. As described in Section 2.2, residual damage cannot be addressed by normal redux actions, and can only be restored with long lane maintenance . We limit the value of the residual to an integer value in the range $[0, 100]$ because any aircraft with residuals 100 or more cannot become FMC by any action other than a complete tear down (long lane maintenance). Therefore, any aircraft with residuals greater than or equal to 100 can be grouped into a state which is not FMC (NFMC) and can only become FMC through long lane maintenance.

If an aircraft is in maintenance, the first value in the state vector represents the SAS number of the aircraft when exiting maintenance and the second, the heavy hitter status of the aircraft exiting maintenance. The third represents the level of residuals the aircraft has when completing maintenance. It is necessary to track all three pieces of information since they all influence the SAS buyback from a given redux package.

The fourth value of the state vector is used to capture the days left of a SAS redux maintenance action. If an aircraft is not in maintenance, this value will be 0.

A state vector with the four aforementioned components has approximately 600,000 different combinations of values. However, by enumerating only feasible states (e.g., a state where we have more residuals that total SAS is infeasible), we can eliminate many of these. Our final state space consists of about 150,000 states, 50,000 of which are flying states.

2.5.2 Actions

In general, if an aircraft is not already in maintenance, it is feasible to assign the aircraft to any SAS redux package or to fly. That is, the SAS state of an aircraft does not limit the set of feasible maintenance actions. For aircraft that are in multi-day SAS redux, the only feasible action in a given time period is to complete the given LO redux package; we do not allow preemption. In particular, if an aircraft is entered into a multi-day SAS redux, we assume it must complete the entire redux package before coming out of maintenance. Operationally this might not be entirely accurate, but it is a fairly realistic assumption. Generally, setup and teardown times as well as scheduling issues between units make it very undesirable and rare to take an aircraft out of maintenance before it is scheduled to do so. Based on this non-preemption assumption, maintenance capacity implications must be considered over the long term. For instance, if the LO maintenance capacity is full, an additional aircraft can not be maintained until another aircraft exits maintenance.

While the maintenance decision is complicated by the choice of multiple SAS redux packages and the maintenance capacity constraint, the decision of which aircraft to fly is a perhaps a little more straightforward. To meet operational requirements in each time period, we are required to provide a given number of aircraft to fly. Based on current fleet sizes, we assume that there are enough aircraft to both fully utilize the SAS maintenance capacity and to fulfill the flying schedule. Thus, some aircraft will neither be maintained nor flown in a given time period and thus the SAS number for these aircraft does not change in that period. For our model, we define seven possible actions:

- Rest (No flying or maintenance actions)
- Fly
- 1 day maintenance
- 2 day maintenance
- 3 day maintenance
- 4 day maintenance
- 11 day maintenance

For an aircraft not currently in maintenance, a decision must be made to either enter the aircraft into maintenance, fly the aircraft, or opt to have the aircraft do neither. For aircraft that are in a maintenance, the only feasible decision is to continue in SAS redux until the prescribed SAS redux package is complete. We use $U(s)$ to denote the feasible set of decisions for given state s.

2.5.3 State Transition Probabilities

The transition probabilities used in our model are both state and decision dependent. In Section 2.3, we presented the empirical distribution of SAS increase. The data used to build the empirical distribution was collected from aircraft that had flown on a given day. If an aircraft did not fly on a given day, then there was no SAS increase data recorded for the aircraft. In our model, we assume that every aircraft that flies experiences a SAS increase drawn from the empirical distribution. Every aircraft that rests experiences no change in SAS.

For aircraft that undergo SAS redux, the transition probabilities depend on the incoming SAS number of the aircraft, whether it not it has a HH, and the amount residuals damage the aircraft has. Due to the large number of possible incoming and outgoing states, we do not have an empirical distribution for the downward transition probabilities. Accordingly, we assume that the expected reduction in SAS number from a maintenance action is a percentage of the incoming SAS number minus the amount of residual SAS. In other words, we assume that our expected SAS reduction is a fixed percentage of the incoming *fixable* SAS and that the actual reduction is uniformly distributed around that expectation (see below). The expected reduction and uniform distribution are based on SAS redux data and discussions with maintenance personnel. Due to setup and tear down time associated with LO redux, we ensure that the expected SAS reduction from a multi-day maintenance action is more than multiple shorter maintenance actions (i.e., the SAS reduction from a 2 day maintenance action is more than two back to back 1 day maintenance actions). Finally, we assume that an aircraft coming out of SAS redux does not have any heavy hitters due to the policy of fixing the most significant damages first.

Table 2.3 shows the percentages used to determine the expected SAS reduction from four of the SAS redux packages. The proportion of SAS number reduction depends on the duration of the maintenance action and whether or not there is a HH. In addition, we increase the SAS reduction percentage for aircraft with an incoming SAS number greater than 175 and a HH. This is due to the fact that canopy damages, which lead to an immediate 150 unit increase in the SAS number, are the most common reason for an aircraft to have a SAS number in the high 100s or 200s. Since the canopy damage can be repaired quickly, the reduction from SAS redux maintenance for aircraft with canopy damage is higher than for aircraft with normal HHs or no HHs. After the expected

Table 2.3: SAS Redux Package Expected Percentage Decrease with Residuals

	MX Duration (days)			
	1	2	3	4
Non-HH	0.3	0.52	0.67	0.77
HH with SAS $\leq$ 175	0.6	0.66	0.71	0.86
HH with SAS $>$ 175	0.95	0.96	0.97	0.99

SAS reduction has been determined using the reduction percentages, the actual SAS reduction is uniformly distributed around ±10% of the expected SAS reduction. For example, if an aircraft entered a 1 day maintenance action with HHs and a fixable SAS of 100, the aircraft will receive a SAS buyback in the interval $[54, 66]$. An 11 day maintenance action, however, corresponds to a complete overhaul of the aircraft's external coating and so we assume the outgoing SAS number is independent of the incoming SAS number and level of residuals. For the 11 day maintenance action, the outgoing SAS number is always uniformly distributed over the interval $[20, 35]$.

If an aircraft is in a non-maintenance state and the decision is made to enter the aircraft into maintenance, it will transition with probability 1 from its current state to the corresponding maintenance state. Similarly, if an aircraft is already in the middle of a multi-day maintenance action, it will transition with probability 1 to the state that corresponds to the next day of the same maintenance action. We denote the transition probabilities $p_u(s, s')$ associated with an initial state s, ending state s' and the decision u.

2.5.4 State Reward Function

The overall objective of the LO maintenance scheduling problem is to maintain a high FMC rate. Therefore, we provide a positive reward for aircraft that have a SAS number below the FMC threshold of 100. We provide no reward for any states that are NFMC or in maintenance, as they do not contribute to a high FMC rate. We let $g(s)$ denote the reward gained from being in state s. We define $g(s)$ as follows:

$$g(s) = \begin{cases} 1 & s \in \text{set of states with SAS number} \leq 100 \text{ and not in maintenance} \\ 0 & s \in \text{set of states with SAS number} > 100 \text{ and not in maintenance} \\ 0 & s \in \text{set of states that represent being in maintenance} \end{cases}$$

Note that we provide a reward based solely on the state of the aircraft and not the action chosen.

2.6 Computing Index Policies

Next we discuss how to adapt the Bertsimas and Niño-Mora approach to derive index policies based on an LP relaxation. This LP heuristic produces two index policies that are used to decide when and how to enter aircraft into SAS redux as well as which aircraft should be flown on a given day to satisfy sortie operations. Based on the historical data, we assume that all aircraft are identical in their dynamics so that the index values $\lambda_s = \lambda_{s_i}, \lambda_{s_{i+1}}, ...$ are only state dependent, not state and aircraft dependent. Once the index values have been calculated for all possible states, aircraft are ranked according to their state dependent index values. The highest ranking aircraft are entered into maintenance so that the maintenance capacity is fully utilized in every period. However, since there are multiple SAS redux packages, we must also choose the redux package to enter an aircraft into. The remaining aircraft are then ranked in terms of which should be flown. The highest ranking aircraft are then chosen to fly for that day.

The decision variables, x_s^u, in this model represent the total expected discounted amount of time the fleet will spend in state s and take action u. Mathematically this would mean

$$x_s^u = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{a=1}^{A} \frac{I_{s,u}^a(t)\cdot\beta^t}{A}\right]$$

where A is the total number of aircraft and

$$I_{s,u}^a(t) = \begin{cases} 1 & \text{if aircraft } a \text{ is in state } s \text{ and action } u \text{ is taken at time } t \\ 0 & \text{otherwise} \end{cases}$$

To generate these indices, the linear program we consider is:

$$\max \sum_{s\in S}\sum_{u\in U(s)} g(s)\cdot x_s^u, \tag{2.1}$$

subject to

$$\sum_{u\in U(s)} x_s^u = \overline{x_s} + \sum_{s'\in S}\sum_{u\in U(s')} \beta\cdot p_u(s's)\cdot x_{s'}^u, \qquad \forall\, s\in S, \tag{2.2}$$

$$\sum_{s\in S_m}\sum_{u\in U(s)} x_s^u \le \frac{\mathcal{M}}{1-\beta}. \tag{2.3}$$

$$\sum_{s\in S_f}\sum_{u\in Flying} x_s^u \ge \frac{\mathcal{F}}{1-\beta}. \tag{2.4}$$

$$x_s^u \ge 0 \tag{2.5}$$

The objective function maximizes the expected reward while constraints (2.2) represent the transition dynamics of the problem. In these constraints, $\overline{x_s}$ represents the initial conditions which can either be the fraction of aircraft in state s at the beginning of the planning horizon, or some distribution over the states. Constraint (2.3) ensures that the total expected discounted fraction of aircraft in maintenance does not exceed the capacity where $\mathcal{M}$ is the repair capacity divided by the total number of aircraft. We define $S_m \subset S$ to be the set of all maintenance states and use $\mathcal{M} = \frac{5}{40}$ which would emulate a base with 5 maintenance bays and 40 aircraft. Next, constraint (2.4) ensures that the chosen plan meets the daily flying sortie requirements. In particular, we ensure that the total expected discounted fraction of aircraft that fly meets the the operational requirement, where $\mathcal{F}$ is the daily sortie requirement divided by the total number of aircraft. We define $S_f \subset S$ to be the set of all non-maintenance states and use $\mathcal{F} = \frac{16}{40}$ which would emulate a base that is required to fly 16 aircraft each day. Finally, β is the discount factor which we set to 0.99. Large increases in SAS number occur with a low probability, but have a significant impact on the objective function. Accordingly, we use a high value of β value to "look" farther into the future to capture these effects.

The model presented above has a few differences from the LP relaxation proposed by Bertsimas and Niño-Mora [6]. First, the original formulation considers non-homogeneous bandits. As mentioned previously, statistical tests indicate that there are no significant factors to differentiate the SAS evolution of different aircraft. Accordingly, we can consider all aircraft to have the same SAS evolution process which leads to a reduction in the number of variables in the formulation. Second, unlike the original formulation, there are multiple ways to "operate" a bandit in the SAS redux problem. Accordingly, we must include variables for each of the possible maintenance decisions and the flight decision. Third, we introduce an operational sortie requirement constraint. This constraint ensures that we satisfy our daily sortie requirements by choosing enough aircraft to fly each day.

2.6.1 Computational Tractability

Although we include only a few of the many pieces of information necessary to get a complete characterization of the SAS state, our state space is quite large. The LP outlined in equations (2.1)-(2.5) remains tractable, but takes a significant amount of time to solve. For this reason we elect to solve it only once, using a starting state, $\overline{x_s}$, that is uniform across the state space, rather than re-solve each time the state of the fleet changes (every day). Using a uniform starting state gives us an offline policy that we can solve with barrier methods in only a few minutes, and does not require re-solving depending on the starting state of the system. In particular, we assign $\overline{x_s}$ values according to the following rule:

$$\overline{x_s} = \frac{1}{|F|} \qquad \forall s \in F \tag{2.6}$$

$$\overline{x_s} = 0 \qquad \forall s \in M \tag{2.7}$$

F is the set of flying states and M is the set of maintenance states. This method assumes that aircraft are equally likely to be in every flying state. In simulation experiments we find that most of the time this model gives us the same index ranking that we would get from solving online (exact values for $\overline{x_s}$) for a given day. Even when there exist differences, there are generally very few, and only between states that have very close indices in both policies. Other methods of choosing $\overline{x_s}$ are outlined in Appendix B, but none of these produced better results.

When we code this LP in AMPL and use the CPLEX barrier method solver (baropt) we obtain a solution in just under ten minutes. Both primal and dual simplex solvers take over two hours, which could be due to degeneracy inherent to the formulation.

2.6.2 Generating Indices

Solving the linear program above yields both a primal and dual solution. We will use the reduced costs of the decision variables to determine the indices. Let γ_s^u be the reduced cost associated with variable x_s^u in the optimal solution. For each non-maintenance state $s \in S_f$, we calculate the maintenance index, λ_s, in the following manner:

$$\lambda_s = \max_{u \in U_{MX}(s)} \gamma_s^u - \gamma_s^{Rest} \tag{2.8}$$

where $U_{MX}(s)$ represents the set of all available actions for state s that involve entering an aircraft into maintenance. The maintenance action associated with state s will be the maintenance action that attains the maximum in (2.8) above. The intuition of this method is fairly straightforward. If we interpret the γ_s^u values as "the amount by which our objective would decrease if x_s^u were included as a basic variable", it is clear that we choose the action that is least detrimental to our objective, meaning the x_s^u which has the greatest (least negative) reduced cost. If the model chooses to take an action rather than rest for a specific state, we can measure how detrimental it would be to our objective to rest instead, by looking at the reduced cost of the x_s^{Rest} variable.

We generate our flying index, μ_s, in a similar manner:

$$\mu_s = \gamma_s^{Fly} - \gamma_s^{Rest} \tag{2.9}$$

These index values allow maintenance schedulers to easily determine which aircraft should be placed in maintenance once capacity becomes available, and which aircraft should fly. In addition to ranking aircraft, these index policies also yield maintenance decisions that determine the SAS redux package an aircraft should be scheduled for. It is important to note that because we use differences in reduced costs from an LP, rather than subsidy values in a DP like Whittle, we are able to generate both indices from one solution of the model. This method could be generalized to accommodate even more types of actions if necessary.

As explained earlier in this section, each day aircraft are ranked using the maintenance index policy developed above. Aircraft are selected for maintenance accordingly, with each aircraft also having a corresponding maintenance package prescribed. The remaining aircraft are then ranked in terms of which should be flown using the flying index policy. The highest ranking aircraft are then chosen to fly for that day. The remaining aircraft rest and experience no change in SAS state.

2.6.3 Policy Characterization

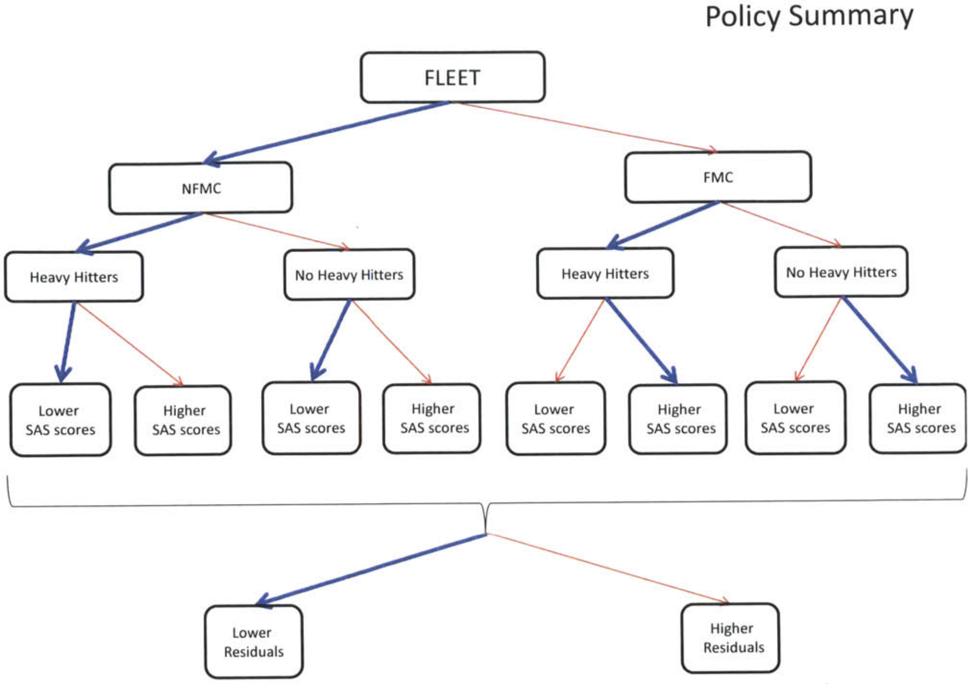
Another benefit to generating an offline policy is that we can easily characterize our index policies in terms of several simple operational rules that they suggest. The general characterization of our maintenance index policy is quite intuitive and simple. As expected, our policy instructs us to maintain aircraft that are not FMC before any aircraft that are FMC. Aircraft that are FMC have negative indices indicating that we lose reward by maintaining them (which we know to be true). We observe very high indices for aircraft whose SAS numbers are just over 100. This is probably due to the fact that these aircraft can be moved into a state that gets a positive reward with minimal use of resources (e.g., an aircraft with a SAS number of 101 is highly likely to become FMC with only a single day of maintenance). We then see a slow decline, indicating that aircraft that are closer to 100 are better to maintain. Higher levels of residuals lower the values of the indices, but the general trends just described generally hold. States with heavy hitters seem to behave similarly, but tend to have higher indices. In general, one could use the following ranking system to approximate the given policy:

1. Maintain NFMC aircraft before FMC aircraft.
2. Maintain aircraft with heavy hitters.
3. Maintain aircraft that have the lowest SAS closest to 100.
4. Maintain aircraft with lower residuals.

This policy is explained visually in Figure 2.6.1. The thick blue line represents the preferred aircraft to work on. All the children nodes of a preferred type of aircraft are preferred to the children nodes of the non-preferred aircraft. For instance, we prefer to work on an NFMC aircraft before an FMC aircraft regardless of any other factors.

For most HH states, our index policies enter an aircraft into a one day maintenance action. This is expected due to the high marginal returns from the first day of SAS redux on an aircraft with a HH. For large SAS numbers, the LP index policy increases the length of maintenance. In

Figure 2.6.1: Visualization of Index Policy Characterization

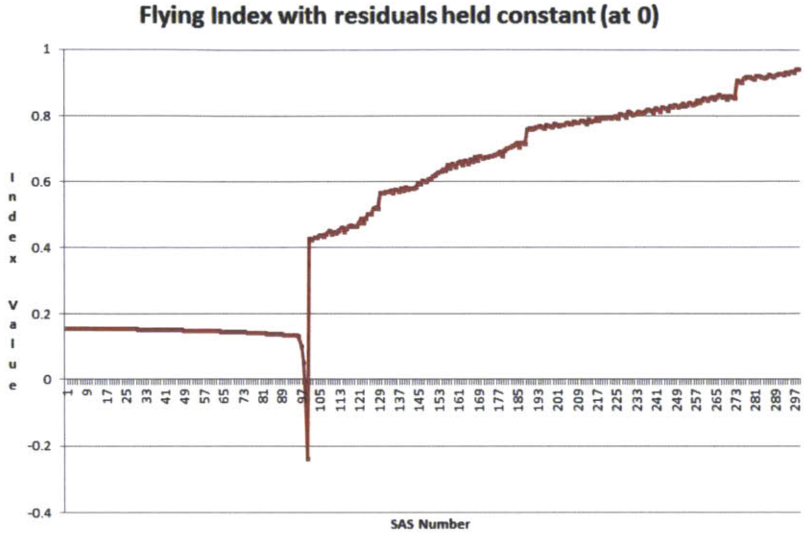


general, as the SAS number increases, the associated maintenance action increases in duration. This follows intuition since an aircraft with a higher SAS number needs to undergo a longer SAS redux maintenance action to return to an FMC state. Finally, long lane maintenance is chosen for aircraft with very high levels of residuals or very high SAS numbers with no heavy hitters. In general the index policy indicates we should put the aircraft with the highest residuals into long lane maintenance.

The characterization of our flying index policy is also intuitive. However, we note that currently there seems to be no awareness among Air Force practitioners to the importance of flying decisions. In general, our policy suggests that we fly aircraft that have already become NFMC first if they cannot be put into maintenance. We then seek to fly the aircraft that have the smallest chance of breaking the FMC threshold, which would be the aircraft with the lowest SAS scores. Figure 2.6.2 shows the flying index values generated by our LP for all states with no heavy hitters. It shows that aircraft with high SAS scores are most valuable to fly, since the aircraft is already NFMC.

The intuition behind this policy is the fact that SAS increases to aircraft that are already above the FMC threshold have little impact in the near term. Due to the high probability of low SAS increases seen in the empirical PMF, the SAS number of these aircraft will likely not increase significantly. In addition, the low SAS numbered aircraft are not likely to cross the FMC threshold. In essence, this is a greedy policy that maximizes the expected FMC rate in the next period.

Figure 2.6.2: Flying indices for the LP index policy when residuals are held at 0 (No HH).

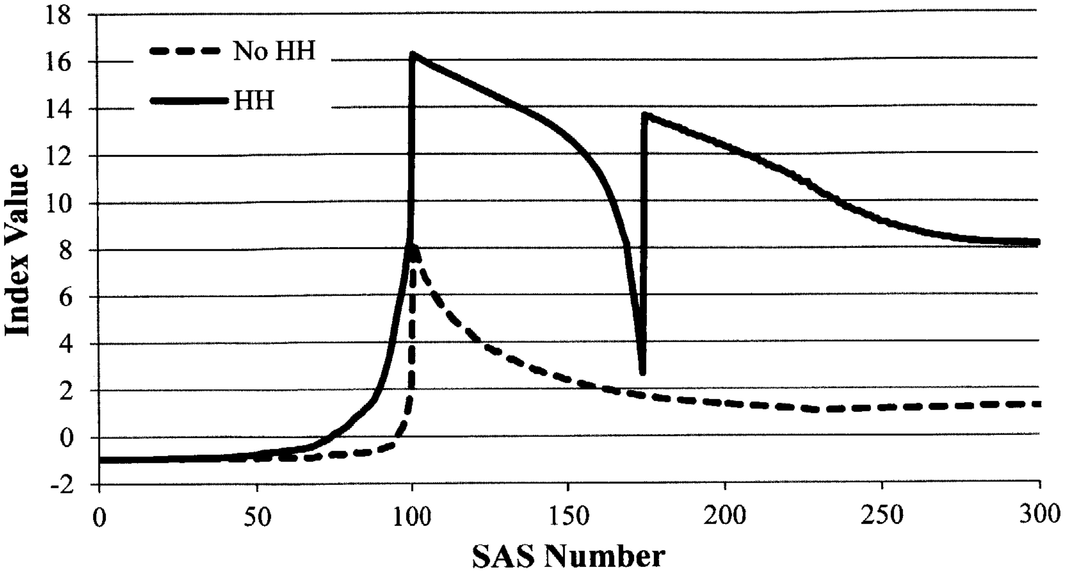


2.6.4 Dynamic Program Index Policy

Finally, for comparison purposes, we also solve a DP which models the LO problem, though less rigorously. Since the DP model is significantly more computationally intensive, we do not include the "residuals" component of our state space. In addition, we do not solve for a flying index, as the Whittle method of solving for indices as subsidies only works to find one index. Since the DP maintenance index policies do not include residuals, they are easy to present visually. Figure 2.6.3 shows the values of the DP indices by state. The DP index policy suggests that 1 day maintenance be performed on all aircraft with heavy hitters. For aircraft with no heavy hitters, the DP policy recommends increasing lengths of maintenance as SAS number increases.

The large dip in Figure 2.6.3 is likely an artifact of the fact that we afford a much greater decease for aircraft with heavy hitters with SAS number above 175, in order to account for canopy strikes. Due to our simplified downward transition probabilities, this policy decides that states with heavy hitters and SAS just below 175 should wait to be maintained, since they will achieve a much higher reduction after reaching a SAS number of 175 (see Table 2.3). For details on this DP generated

Figure 2.6.3: DP Index values associated with each possible SAS state.



policy please refer to Appendix A.

Overall, the index values generated by the linear program are similar in nature to those generated by the dynamic program. Both exhibit a jump around 100 and general preference of heavy hitters over non heavy hitters.

2.7 Simulation

In order to establish the efficacy of our policies, we test them in an environment that mirrors reality as closely as possible. In this section, we describe a simulation setting that is used to assess, at least approximately, how the proposed policies might perform in practice compared to current policies and other possible heuristics. The simulation is based on descriptions of operations from the same Air Force flying unit that provided the SAS creep data. Each trial has a planning horizon of 1000 days. On each day 16 of the 40 simulated aircraft are required to fly. Aircraft that fly are subject to SAS increases that follow the empirical distribution presented in Section 2.3. Aircraft that do not fly are assumed to have no change in SAS number. Lastly, we assume an LO maintenance capacity of four normal redux bays and one long lane bay. We assume the long lane maintenance bay is the only bay that can perform long lane maintenance, because it is the only location with the necessary set-up and equipment.

Table 2.4: Redux Bay Availability Probabilities

Number of Bays Available	**Probability**
0	8%
1	27%
2	30%
3	15%
4	20%

In each trial of the simulation, we specify a maintenance policy that is used to determine which aircraft to maintain and what type of maintenance to perform. After deciding which aircraft to maintain, a flight decision policy is used to select which of the remaining aircraft will fly. Although we did not include FOM implicitly in our MAB model, we do include it in our simulation setting to ensure that our policies remain valid in an environment where FOM maintenance occurs. As described in Section 2.2, FOM maintenance occurs when panels are removed in order to access internal components of the aircraft for maintenance purposes. LO maintainers must replace these panels once internal maintenance is complete and restore the LO coating. FOM's major impact comes in the form of maintenance resource availability. In reality, there are days where there are no maintenance bays available for redux maintenance, due to a large demand for FOM. Using maintenance data from the same flying unit that provided the SAS creep data we can make a rough estimation of the impact of FOM on bay availability. These estimated values are shown in Table 2.4. Essentially, we see that although there are 4 standard maintenance bays, we usually can perform redux on only 2 or 3 aircraft, sometimes fewer.

Our simulation also allows us to see how our policies behave in an environment where more is known about the distribution of damages for each aircraft. In reality, when an aircraft enters maintenance, maintainers identify the damages that contribute most to the SAS number of the aircraft, and focus their efforts towards those damages. In our models, keeping track of each damage on each aircraft would quickly cause the model to become intractable, which is why we elect to use the downward transition probabilities described in Section 2.5.3 instead. However, our simulation can operate like the maintenance squadrons do in reality meaning that we can track the distribution of damages for each aircraft. Once we choose an aircraft and a maintenance package, maintenance is performed according to the distribution of SAS damages for the given aircraft. The largest damages are addressed first, and the number of damages addressed each day is uniformly distributed between 3 and 5.

2.7.1 Other Maintenance and Flying Heuristics and Policies

In addition to the proposed index policy, we include in the computational experiments a simplified version of the current maintenance policy, implemented in the fighter squadrons that provided the data in Section 2.3. We simulate this policy by fixing all aircraft with a heavy hitter immediately with a one day maintenance, and then choosing randomly from the remaining aircraft. Although this policy is not exactly what is done in practice, in provides a good benchmark against which we can test the performance of our policies. We entitle this policy the *naive policy.*

In addition to our flying LP index policy, we also explore a few general heuristics that can govern which aircraft to fly on a given day. These heuristics consider the SAS creep and maintenance processes and can be implemented in the collaborative environment. Discussions with maintenance personnel indicate that flight decisions are currently made at random with regards to LO status. That is, of the aircraft that are currently available to fly, each one is equally like to be selected for flight. We refer to this policy as random selection.

Based on the long term goal to maintain a high FMC rate, we consider the following flight assignment heuristics:

1. random selection,
2. high,
3. low,
4. high-low.

For a flight schedule requiring D aircraft, the random policy selects D of the aircraft not currently in maintenance, each with equal probability. The High and Low policies each select D consecutive aircraft with the former selecting the D highest SAS numbered aircraft and the latter the D lowest SAS numbered aircraft.

The fourth policy that we test combines these two policies based on the FMC threshold. If D aircraft have SAS numbers above the FMC threshold, the high-low policy will select the D highest aircraft. If not, it will select all of the aircraft above the threshold and will select the lowest

SAS numbered aircraft for the remainder. For example, for the current FMC threshold of 100, an available fleet of 5 aircraft with SAS numbers of $\{23, 39, 86, 102, 167\}$ and $D = 4$, the high-low policy will first select the two aircraft above the threshold, $\{102, 167\}$, and then the two aircraft with the lowest SAS numbers, $\{23, 39\}$. The intuition behind this policy is the same as the index that we developed with our LP and characterized in Section 2.6.3. In the subsequent section, the quality of these policies will be explored through simulation.

2.7.2 Performance

Using the simulation as detailed above, we test the performance of our index policies and the other policies. The results can be viewed in Figures 2.7.1 and 2.7.2. The legend indicates which policies are used, giving first the maintenance policy, then the flying policy. We begin our analysis of the results by observing the performance of the naive maintenance policy combined with the random flying policy. This combination is our best approximation of the way LO decisions are currently made. Our results predict an FMC rate between 60% and 70%, which roughly matches some of the units that were having trouble keeping LO FMC rates up in 2011. This leads us to believe that our simulation is accurately reflecting the scenario, since our results match reality.

Next, we view the performance of the naive maintenance policy combined with the high-low flying heuristic. Remarkably, we see that the addition of a good flying rule adds almost 20 percentage points to the FMC rate. This indicates that integrated decisions could go a long way towards improving over FMC rates.

It is interesting to note that by adding a good flying rule, our FMC rate jumps by 20 points, whereas when we add a good maintenance policy (such as our LP index policy) and leave flying random, we see an increase of only about 15 percentage points. This indicates that flying policies could be as important as maintenance policies.

Finally, we turn to the performance of our index policies with and without flying rules. We see that our LP generated index policies tend to outperform the DP index policies both with and without a flying rule, and they both outperform the naive policy. The difference between the LP policy and DP policy is likely due to the fact that the LP has an increased state space that allows it to account for residuals, whereas the DP uses only total SAS score and heavy hitter status. As with the naive policy, we see a large increase in FMC rate of about 15 to 20 percentage points when we add a flying rule. The "high" and "low" flying heuristics described in Section 2.7.1 produced FMC rates that were worse than the high low heuristic and are therefore not included in Figure 2.7.1.

In summary, these results provide the following insights

1. A good flying rule, such as the high-low heuristic or our flying index policy, can greatly increase FMC rates for a fleet.

2. Since the high-low flying rule and the flight index policy perform similarly, it appears that heavy hitter status and residual levels are not very important in making flight decisions. Knowing the total SAS score of each aircraft is enough to make good flight decisions.

3. Both the LP and DP index policies outperform the naive policy, indicating that by prioritizing maintenance actions according to the rules outlined in Sections 2.6.4 and 2.6.3, FMC rates could be increased.

4. Policies that account for residuals do better than those that don't, indicating that it is important to include residuals when making maintenance decisions.

2.7.3 Sensitivity analysis

In the previous section we showed that our index policies work well for the scenario descried by the maintenance unit and the associated data. In this section we seek to alter the scenario to see if our index policies perform robustly. Specifically we will examine what happens when we alter the total number of normal maintenance bays (not long lanes), and the size of residuals (since these are only roughly estimated). Tables 2.5 through 2.8 show average FMC rate over a 1000 day horizon with varying levels of maintenance bays and residuals, for different policies.

These results clearly indicate a few trends. First, we see that using a flying rule continues to improve the fleet FMC rate by a large amount, even when we have low MX capacity or high residuals. Although we know choices concerning which aircraft fly on a given day will probably never be based on LO concerns in practice, it is interesting to see what a large difference a flying policy that incorporates LO status considerations can make, and how much more robust it can make the FMC rates. Next, we see that as maintenance bays increase, the performance of all policies tend to converge to common values. In other words, as we gain more maintenance bays, we eventually reach

Table 2.5: Performance of LP index policies with variable residuals and MX bays

Number of Bays/ Resid level	0.1	0.3	0.5	0.9
3	76.0%	75.7%	74.2%	72.3%
4	94.7%	90.8%	84.7%	76.9%
5	97%	95.4%	88.7%	77.7%
6	97.5%	96.7%	90%	79%

Table 2.6: Performance of DP index policy and "High-Low" flying heuristic with variable residuals and MX bays

Number of Bays/ Resid level	0.1	0.3	0.5	0.9
3	75%	74.5%	71.6%	69%
4	93%	87.9%	80.8%	72.1%
5	97%	96.1%	88.4%	74%
6	96.8%	96.8%	91%	74.8%

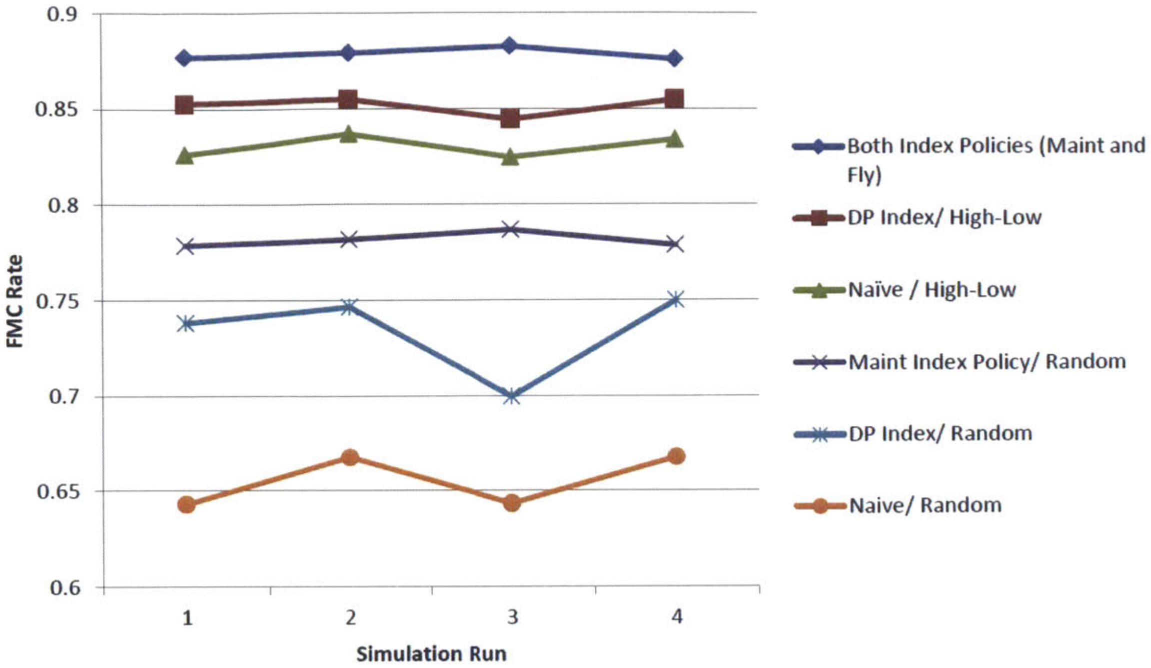
Table 2.7: Performance of Current policy and "High-Low" flying heuristic with variable residuals and MX bays

Number of Bays/ Resid level	0.1	0.3	0.5	0.9
3	73.8%	72.3%	69.3%	67%
4	88%	84.4%	80.9%	72%
5	96.4%	95.2%	88.3%	75.2%
6	97%	96.7%	90%	75.8%

Table 2.8: Performance of naive policy with random flying policy with variable residuals and MX bays

Number of Bays/ Resid level	0.1	0.3	0.5	0.9
3	45.4%	39.9%	30.9%	26%
4	75.7%	69.4%	57.3%	40.3%
5	93.8%	89.5%	77.5%	48.3%
6	96.1%	94.5%	82.1%	51.8%

Figure 2.7.1: FMC rates for 4, 1000 period, trials for varying maintenance and aircraft flight selection policies.

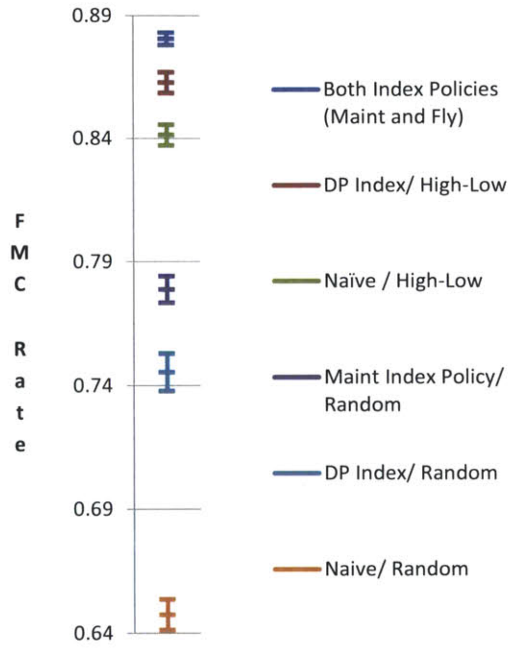


the point where any reasonable selection policy gives us an FMC rate in the high 90's. We also see that decreasing the number of maintenance bays affects all policies similarly, except for the current policy with the random selection flying rule. Finally, as we might expect, we see that as residuals increase the difference between the LP policies (which account for residuals) and the DP policy combined with the high-low rule (which doesn't) increases. With very small residuals, all policies do similarly, whereas we observe the LP policies performing much better with high residuals. This is potentially important since as the fleet ages, the magnitude of residuals could increase.

2.8 Conclusions and Future Work

In this chapter we have demonstrated the power of index based policies applied to the LO maintenance scheduling problem. Our index policies perform very successfully under realistic scenarios that are based on empirical data. Even if for some operational reason the index policies developed here could not be implemented in reality, the insights behind them presented in Section 2.6.3 provide intuition into which aircraft should be maintained, and which should not. These insights could

Figure 2.7.2: Estimated Policy Performance Averages with 95% Confidence intervals



also be incorporated into some other potentially more complicated policy. In addition, we show empirically that good flying rules can be very beneficial to FMC rates. Currently this is something that Air Force practitioners do not consider. Although it might be difficult to implement, including LO considerations in scheduling which aircraft fly on a daily basis could prove very helpful to fleet FMC rates. We also note in Section 2.7.3 that increased levels of capacity tend to nullify any of the benefits that come along with good flying or maintenance policies. Recently, many F-22 bases have added more LO maintenance resources to address low FMC rates. These bases have experienced increases in LO FMC rates since adding these extra resources as our sensitivity analysis predicts.

Areas for future work include time varying sortie demands, the inclusion of other types of maintenance, and the inclusion of FOM in our LP model. The first of these can help further current discussions with the Air Force surrounding different sortie scheduling rules (i.e., fly 14 aircraft on Monday, 12 on Tuesday, etc. versus 10 every day) while the second might help find efficient ways that different maintenance groups might work together when an aircraft is down for any type of maintenance. In addition, we currently view FOM as an action that diminishes the number of bays available to us for redux, since FOM currently takes priority over SAS redux actions. In reality, when FOM is completed, some amount of SAS redux is completed if the damages are readily accessible from the panel being repaired. By including FOM scheduling in the maintenance decision process, the decision to perform SAS redux in addition to FOM could be considered as well as the option to delay FOM in lieu of SAS redux.

From a modeling standpoint, the formulation presented could be expanded if more detailed data were available. In particular, with more data regarding the distribution of damages, the state space could be expanded to more accurately capture the relationship between the distribution of damages and the transition probabilities, specifically the effectiveness of SAS redux. The tractability of the resulting problem must be considered as a significant increase in the state space, and could make the resulting models intractable.

Chapter 3

Fighter Engine Maintenance

3.1 Introduction

In this chapter we turn our focus to another area of maintenance altogether. Unlike LO maintenance, which is concerned with the exterior of the aircraft, maintainers must also perform extensive maintenance work on interior components of a fighter aircraft. Perhaps the most important among these components is the engine. This section discusses some related aspects of how the United States Air Force currently maintains its stock of working fighter aircraft engines and identify challenges and potential areas of improvement.

While there has been a lot of attention among academics and practitioners regarding the study of this complex system, most of the focus has been on availability of spare parts that are indeed essential for the smooth operations of the fighter aircraft. For example, Jackson et al. [16] develop real-time models concerning spare parts resupply methods to ensure necessary parts are available when needed for maintenance. In addition, Muckstadt [20] develops a model to establish efficient stock levels for the echelon structure of the parts resupply process.

However, over the last decade it has become increasingly apparent that maintenance resource allocation is an equally important enabler to the mission readiness of fighter aircraft and should be considered together with inventory issues. That said, there has been relatively little work on these challenges both in the academic literature as well as among practitioners.

This chapter attempts to model and analyze some of the challenges around the scheduling and utilization of maintenance resources. Specifically, we model the maintenance operations of the F100-

229 engine, which is used in the F-15E and F-16C/D aircraft. The models we present capture the dynamic allocation of resources to manage both elective (scheduled) and non-elective (unscheduled) maintenance needs for this system. Each operational base has an Air Force prescribed number of serviceable engines that must be in stock at all times, known as war ready engines (WRE). Whether or not a maintenance shop is able to fulfill its WRE requirement is one of the most important metrics upon which it is evaluated. Generally speaking, the models we develop aim to reveal when and how engines should be maintained in order to meet WRE goals as much as possible.

We begin by presenting a detailed background of the F100-229 engine maintenance process and explain the decisions involved in Section 3.2. We then show how to model the engine maintenance operations that occur at a single base as a variant of the well known restless multi-armed bandit superprocess model in Section 3.4. In particular, the variant we propose includes multiple resources, which are consumed in different combinations depending on the actions taken. The model is based on data obtained from Air Force databases concerning the historical flight hours and breakages of the F100-229 reported over the past 7 years. We supplement this data with qualitative information obtained through conversations with active duty maintenance personnel and program managers. When solved, the model generates indices for each action/ state pair, (a, s), which can be thought of as "the value gained by performing action a on an aircraft in state s". We characterize these indices in Section 3.5. However, since there now exist multiple resources, we cannot simply choose the m highest ranked aircraft and enter them into maintenance. For this reason, we propose two index heuristics which use the indices generated from the proposed model to make maintenance decisions. The first index heuristic uses the sign of each index to determine whether or not an action should be performed. The second heuristic disregards sign, but instead only considers broken engines for maintenance.

Finally, in Section 3.6, we present data-driven simulation results that test the performance of our index heuristics against a naive policy. The results demonstrate the effectiveness of the proposed index heuristics under reasonably realistic scenarios using real data. In particular, the results indicate that the proposed index heuristics substantially outperform naive policies. Specifically, when we compare the performance of a naive policy against the performance of the index heuristics we observe a decrease in the average number of unserviceable engines by almost 20%, a large decrease a decrease in variability, and a 98% decrease in the proportion of days where WRE requirements are unmet.

3.2 Background

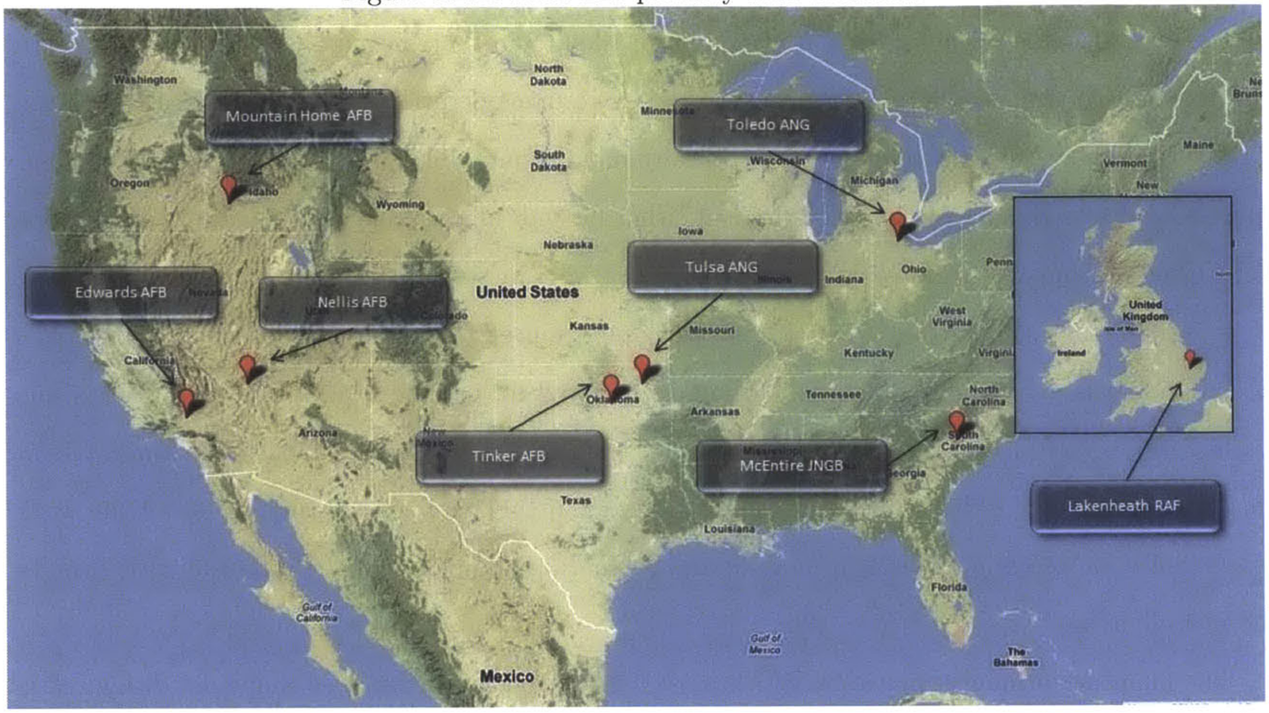
To motivate the modeling work of the fighter aircraft engine maintenance logistics infrastructure, we next describe in detail the maintenance process and the various decisions being made throughout the process. The F100-229 engine is the primary propulsion system used in the F-15E and the F-16C/D fighter aircraft. These engines exist primarily in 3 types of locations: operational bases, central depot , and contract field team locations. The first type of location includes all the bases that house fighter wings of the F-15E or F-16C/D fighter aircraft. Seven such bases exist, six of which are in the continental United States, and one of which is in England. Each of these bases contain several engines and a maintenance facility for performing basic maintenance activities known as a *backshop.* Figure 3.2.1 shows the physical layout of the primary F100-229 bases.

The second type of location is known as the *depot* location. The depot is used as a centralized maintenance facility that performs all extensive repair activities and overhauls. No flying operations occur at this location. Generally speaking, each type of engine has a single depot location. The depot location for the F100-229 engine is Tinker AFB which is also shown in Figure 3.2.1.

Finally, the third type of location is known as a *contract field team* (or CFT) location. Contract Field Teams are hired by the Air Force to perform any extra maintenance that operational bases need to offload. CFTs have a contract negotiated at the beginning of the fiscal year that determines the amount of work they will perform over the coming year. CFTs will perform up to, but no more than, the amount of work detailed in this contract and any unused capacity is lost. Although there are other locations that host operations that require the use of the F100-229 engine, these locations usually have fewer engines, and very limited local maintenance capabilities.

Rather than viewing all of these bases as separate entities that each have their own operational and maintenance requirements, the Air Force views them as nodes in a network that can share engines or maintenance capacity as the need arises. Ideally, each base should be able to satisfy its own sortie and maintenance requirements, however large spikes in demand due to random spikes in breakages or deployment necessities can make the sharing of engines and maintenance resources crucial to overall operational success.

Figure 3.2.1: F100-229 primary base locations



Each engine type in the Air Force inventory has a *network manager* who looks at the total number of working and broken engines at each base and for the Air Force as a whole. It is the network manager's job to help bases cooperate in terms of sharing workload and resources when necessary to ensure the network as a whole is keeping up with demand and fixing enough engines to maintain established levels of spares. Current policy involving engine or resource sharing is ad-hoc, and generally only involves movement when a base is having difficulty keeping up with its workload. In addition, the network manager is responsible for allocating major inventory components (such as the modules described below) amongst the operational bases. Network managers use an integrated computer database that allows them to have real time visibility of performance, workload, resources, and inventory for each node in the maintenance network.

3.2.1 Maintenance Operations

The F100 engine was designed in a "modular" fashion to make engine maintenance easier and more efficient. Each engine is primarily composed of 5 separate large modules: The Inlet Fan, Core, Low Pressure Turbine, Augmentor, and Gearbox. Although there are a few other parts and components included in finished engines, these 5 modules comprise the large majority of the engine. Modules are fully interchangeable, meaning that if a module breaks it can be removed and replaced by a spare working module of the same type, and the engine can resume operations. This minimizes down time of the engines because it prevents entire engines from being unusable if a single component is broken. Spare modules to replace broken modules can either be kept in inventory at the maintenance shops or be ordered from the depot location at Tinker AFB.

It is important to note that when a broken engine is moved to a different base to alleviate issues in demand, it is generally shipped as a whole as opposed to just the broken module. This is due to the fact that often the teardown, buildup, and testing stages of maintenance take a non-trivial amount of time. According to active duty maintainers, if they had enough time to do all the work associated with accessing a broken module and rebuilding after fixing, they might as well just fix the module as well.

The main performance metric by which most fighter engine maintenance shops are evaluated is the number of spare, serviceable engines it has in inventory. These spare, serviceable engines are called "War Ready Engines" (or WRE), and each engine type that is in service in the Air Force inventory has a predetermined number that must be available at any given time (called required WRE). There is a required WRE value for each base and for the Air Force as a whole. The logic behind this is to have enough spare engines available to support a temporary surge in flying operations if required, due to unpredicted wartime necessity. Meeting authorized WRE is one of the top priorities of any engine maintenance shop, and is the primary motivator of the network manager when deciding when to move engines or resources between bases.

Clearly, to meet authorized WRE, maintainers must perform maintenance on engines as they break or deteriorate. There are two general types of engine maintenance activities: Scheduled and Unscheduled. Scheduled maintenance deals with predictable module lifetime. In particular, each module has a pre-determined life limit that specifies a frequency of maintenance that is determined by time (e.g., every 12 months) or usage (e.g., every 2000 flying hours). In contrast, unscheduled maintenance deals with all unplanned or unexpected breakages that occur in a rather random fashion.

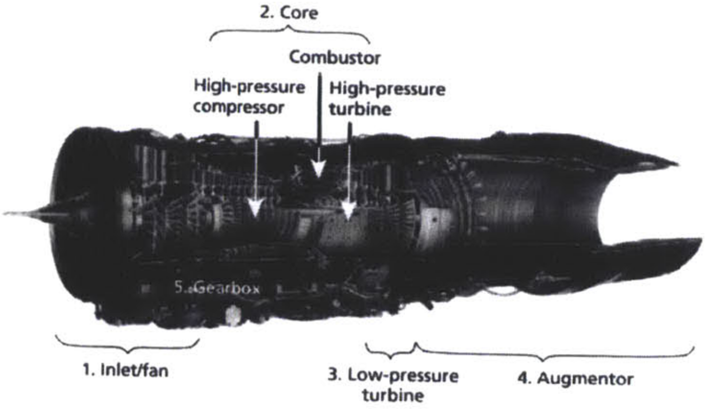


Figure 3.2.2: Assembled F100 Engine with labeled modules [31]

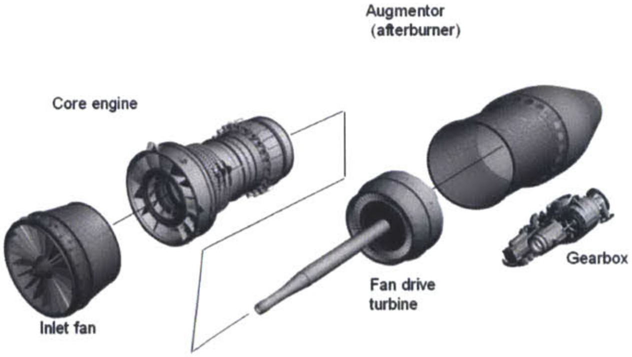


Figure 3.2.3: Disassembled F100 engine [32]

3.2.1.1 Unscheduled Maintenance

Unscheduled maintenance is performed anytime a breakage occurs. When a problem is brought to the attention of the maintenance squadron, they first attempt to fix the engine while it is still installed on the aircraft. Using portable diagnostic equipment maintainers can attempt to find the causes of the reported problems and rectify them while the aircraft is still on the flight line. These maintenance actions are called *flight line* maintenance actions. Flight line maintenance actions are in general relatively simple, and don't take much time to resolve, as they deal only with the exterior portion on the engine. This is because the engine as a whole stays in tact rather than being torn apart. Once an engine is torn apart it will generally take at least 13 days before the engine is operational again, whereas flight line maintenance can be completed in a few hours.

If an engine is found to have damage that cannot be fixed on the flight line, the engine will be removed and replaced with a working engine from the WRE stock. The broken engine is then thoroughly inspected for any type of damage that would make testing dangerous or counter productive (e.g., running an engine with loose debris inside can cause many more things to break). If the engine is determined to be able to run safely, it is placed on a "test cell" where is it attached to a battery of diagnostic equipment and operated. Using the diagnostic equipment, the problem can be narrowed down to a specific module. The engine is then broken down and the broken module is addressed. If damage is found that precludes initial testing, the engine is immediately torn down and the damage fixed. The engine is then built back up and tested in the test cell. If further damage exists the engine will be torn down again and the broken module addressed.

During module maintenance, maintainers have two options: *RRR maintenance* and *RR maintenance*. RRR maintenance stands for, "Remove, Repair, and Replace". In this type of maintenance the broken module is removed and serviced by maintainers in the local maintenance shop called the "backshop". The module is then placed back in the engine. RRR maintenance is generally performed if the maintenance actions required are within the ability of the backshop maintainers.

RR maintenance stands for "Remove and Replace". In RR maintenance, the broken module is not repaired immediately. Instead, a spare serviceable module is taken from the inventory and used to replace the broken module. The broken module is then worked on in-house within the backshop when time allows (rare) or sent to an external centralized depot to be fixed (more common). Usually

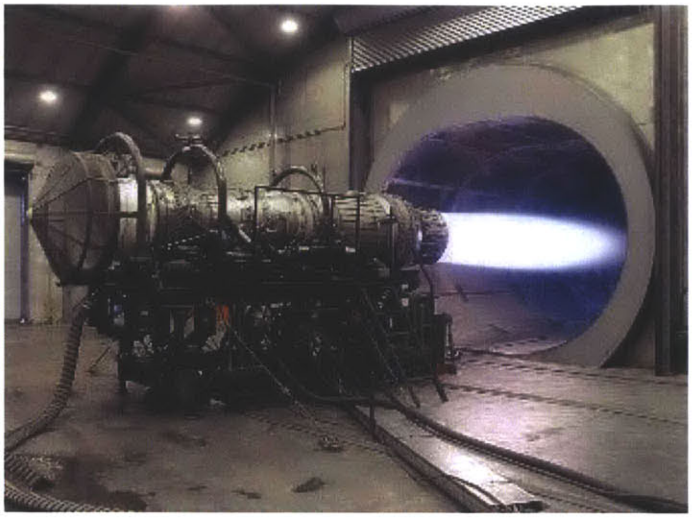


Figure 3.2.4: Test Cell [33]

RR maintenance is performed if a module is too badly broken to be fixed locally. To support this type of maintenance, bases try keep spare modules on hand in inventory. This level of inventory is mainly controlled by the network manager, who decides where to send modules based on forecasted demand. If no modules are available at a base to support RR maintenance, new modules must be ordered from the depot or shipped from another base, which can take several days. It is therefore important to manage the modules in inventory efficiently in order to avoid situations where a module is needed, but there are none available. After either RR or RRR maintenance, the engine is put back together in a process called "build-up" and brought back to the test cell. If problems still exist, the engine is torn down again and fixed using either RR or RRR maintenance. If the engine runs without any further technical issue, it is placed in the WRE stock. Figure 3.2.5 shows the decision tree guiding Unscheduled MX.

It is important to note that many of the decisions described above are not always driven solely by maintenance considerations, per se. For instance, if an unscheduled break occurs on an engine that can be fixed on the flight line but is scheduled to fly the next day, those in command of flight operations might push to have the engine removed and replaced with a serviceable engine so that flying operations can continue the next day without disruption to the flight schedule. This is driven by the performance metrics of the flying squadron that are focused on minimizing flight disruptions and cancellations. However, this is suboptimal for the maintenance squadron, since it diminishes the WRE stock unnecessarily, as well as introduces excess work (uninstalling old engine, reinstalling

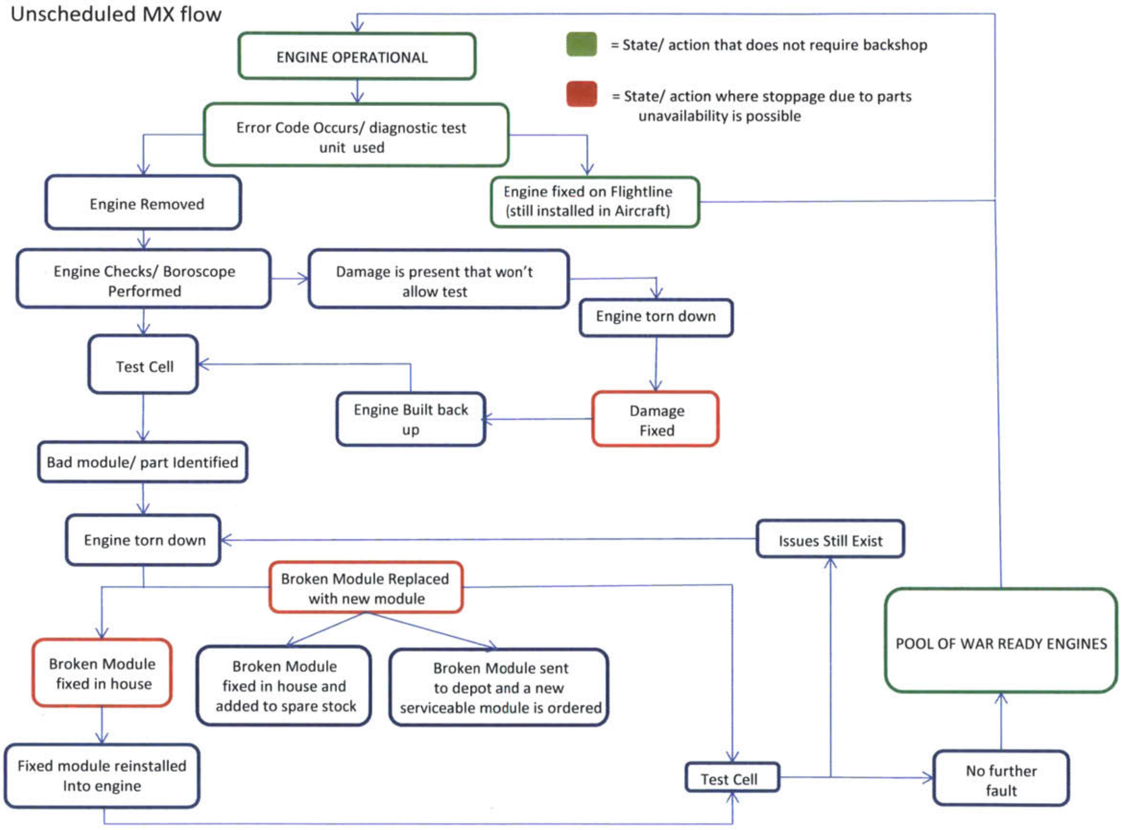

Unscheduled MX flow
ENGINE OPERATIONAL
🟩 = State/ action that does not require backshop
🟥 = State/ action where stoppage due to parts unavailability is possible
Error Code Occurs/ diagnostic test unit used
Engine Removed
Engine fixed on Flightline (still installed in Aircraft)
Engine Checks/ Boroscope Performed
Damage is present that won't allow test
Engine torn down
Test Cell
Engine Built back up
Damage Fixed
Bad module/ part Identified
Engine torn down
Issues Still Exist
Broken Module Replaced with new module
POOL OF WAR READY ENGINES
Broken Module fixed in house
Broken Module fixed in house and added to spare stock
Broken Module sent to depot and a new serviceable module is ordered
Fixed module reinstalled Into engine
Test Cell
No further fault

Figure 3.2.5: Unscheduled MX Flow

new engine *and* maintenance, as opposed to just performing maintenance).

3.2.1.2 Scheduled Maintenance

The second type of maintenance activities that must be performed on engines is called *scheduled maintenance*. For safety and reliability reasons, every module has a predetermined operating life. This life is quantified in either operating hours (as is the case for gearbox and augmentor modules), or *cycles* (as is the case for the inlet fan, core, and low pressure turbine modules). Cycles allow the inclusion of throttle settings into module lifetimes, as opposed to basing them solely on hours. However, discussions with Air Force practitioners indicate that all modules of the F100-229 have approximately 2000 hours of operational life, regardless of how their remaining life is tracked.

If a module reaches the end of this operating life it is considered unserviceable and can no longer be used. Operating life can be restored by undergoing a complete overhaul, during which the module is inspected and fixed from any wear and tear due to flying operations. This overhaul can be completed only at the centralized depot location, where the necessary equipment and expertise exists. If an engine has a module that is approaching the end of its serviceable life, maintainers will order a new module from the depot. They then decide a convenient time to remove the engine for maintenance. Upon removal, the engine is inspected for any additional damage or wear/ tear that can be fixed while the engine is down. If any is found, it is addressed. The old module is then swapped out for the new module, the engine built back up, and the completed engine brought to the test cell. If the engine runs smoothly without ay issue, it is added to the WRE stock and the old module is sent back to the depot for overhaul. This type of maintenance resembles the RR maintenance described above very closely. Since there are no broken parts involved in scheduled maintenance (given that no additional damage is found when inspected), it generally doesn't take as long as RRR maintenance, as long as there is a new module available. The scheduled maintenance flow is outlined in Figure 3.2.6.

Both scheduled and unscheduled maintenance require the use of the same maintenance resources. In particular, both types of maintenance require manpower and a supportable rail, which is the "stand" that the engine is placed on when it requires maintenance. Although there are many other necessary maintenance resources, discussions with Air Force practitioners indicate that supportable rails and manpower are the two limiting factors in accomplishing engine maintenance activities.

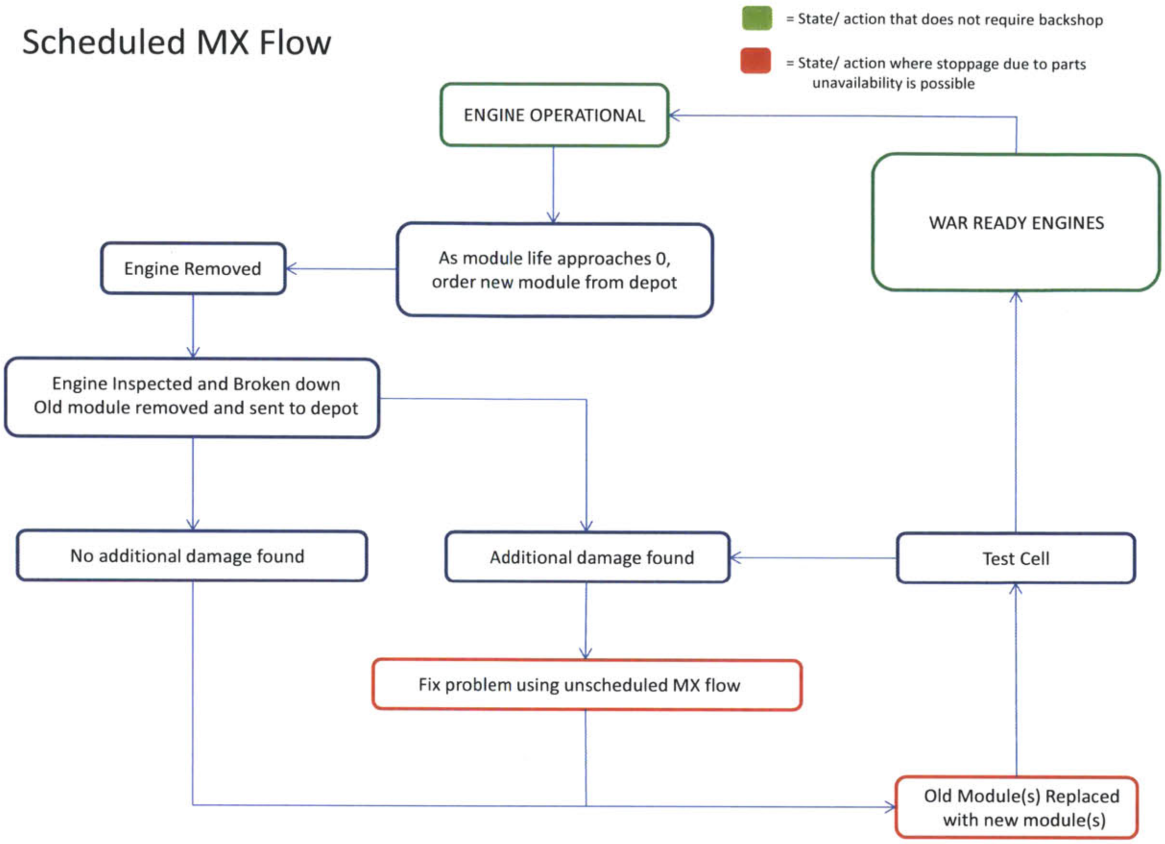

Scheduled MX Flow
🟩 = State/ action that does not require backshop
🟥 = State/ action where stoppage due to parts unavailability is possible
ENGINE OPERATIONAL
As module life approaches 0, order new module from depot
Engine Removed
Engine Inspected and Broken down Old module removed and sent to depot
No additional damage found
Additional damage found
Fix problem using unscheduled MX flow
Old Module(s) Replaced with new module(s)
Test Cell
WAR READY ENGINES

Figure 3.2.6: Scheduled Maintenance Flow

Current Air Force practice is to keep the remaining life of each module around the same value so that when the engine is removed for scheduled maintenance, all modules can be replaced and sent back to depot at once. The practice of keeping all module lives similar is called *alignment*. Alignment minimizes the number of times that a given engine will need to be removed from an aircraft for scheduled maintenance. In particular, if remaining lifetimes were not taken into account when placing modules in engines, this would necessitate tearing down the engine for scheduled maintenance every time a module runs out of life. This will adversely affect the engine up-time and significantly impact workloads for maintainers. By keeping the modules aligned, maintainers can replace multiple modules each time an engine is torn down for scheduled maintenance.

On the other hand, although it would be possible to overhaul modules that have significant amounts of remaining life to avoid frequent removals, this would clearly be expensive and inefficient.

Because of this, there are Air Force regulations for the maximum remaining life allowable before overhaul. For example, if the core module has more than 100 flying hours remaining, the depot will not accept it for overhaul (it will, however, accept it if there exists a damage that cannot be repaired at the local backshop). This is the primary reason that maintainers attempt to perform RRR maintenance on unscheduled breaks as opposed to RR maintenance. When RRR maintenance is performed the module that is getting fixed has no change in remaining lifetime, which allows all modules to remain aligned.

3.3 Modeling Framework

The engine maintenance challenges described in Section 3.2 are extremely intricate and complex. For this reason we must make some assumptions and limit our scope so that our model can remain tractable.

First, we choose to focus our efforts on the maintenance actions performed at a single base. This will significantly decrease the size of our problem, and will provide insight into how a base might operate efficiently if it cannot make use of the other backshops in the network. In addition, we will only focus our attention to the actions occurring in the backshop, meaning we do not account for flight line maintenance. However, this should not make the model invalid, as flight line maintenance generally doesn't take much time, and for the most part only uses manpower resources. It, by definition, will never use resources in the maintenance bay or modules from inventory.

Second, we will assume that every module that is removed is immediately sent to the depot. We will also assume that every new module that is installed into an engine is new and has a full working life. This eliminates the option of removing modules and working on them afterward and the possibility of swapping modules between the engines or with modules in the inventory to get a better alignment of modules lives. Although this is not entirely realistic, it makes our analysis far more tractable. We hope to add these options in future work.

Fourth, we assume that all engines act identically. This means that no engine will have a higher probability of breakage than any other. Since engines are essentially just a combination of modules which often change, this is a very realistic assumption.

Fifth, we assume that only one module may experience an unscheduled break at a time. This is generally true, as if one module breaks the engine is immediately taken out of service, and no other module has time to break. Although some catastrophic failures might cause more than one module to break at once, it is safe to assume that this is quite rare. This means that only one module per engine can undergo repairs at one time. It is important to note that multiple modules can run out of serviceable life at the same time though, and modules can run out of life at the same time as other modules are breaking.

We also assume that maintenance shops use a base-stock policy for their module inventory. Although this is not entirely accurate, the fact that network managers are constantly adjusting which bases receive new modules according to forecasts would make exact characterization of the current policy extremely difficult. In addition, most research in this area assumes base stock type inventory policies.

One possible way to model the engine maintenance problem described above is through a dynamic programming (DP) formulation. However, as in the LO maintenance case, the resulting formulation is again not likely to be tractable. In particular, just to model the maintenance capacity constraint within a DP framework, the state of the system would need to include information regarding every F100 engine at the base. In addition, the state will have to keep track of both on hand and pipeline inventory. For each engine, information about the amount of remaining lifetime for each module, as well as whether or not each of its modules is experiencing an unscheduled break would be necessary. Given that, in practice, there can be over 100 F100-229 engines at a single base and module lifetimes can take on a wide range of values, the size of the state space necessary to capture the relevant information about the system would easily make the DP intractable. In addition, even if the DP could be solved optimally it is not likely to devise policies that will be operationally simple or realistic to implement in practice.

Due to the issues involved in modeling this problem as a dynamic program, we are motivated to find another framework and modeling approach to study the challenges that arise in the context of scheduled and unscheduled maintenance and devise effective operational policies for generating a engine maintenance schedule. We present the multi-armed bandit (MAB) problem as a tractable alternative to a DP. We provide detailed literature review on the various multi-armed bandit models and policies that were studied in the past in Section 2.4 .

3.4 MAB Formulation

In this section, we discuss how to model the engine maintenance scheduling problem as a generalized restless MAB superprocess with multiple resources. The engine maintenance scheduling problem resembles the construct of the MAB superprocess problem discussed in chapter 2 for the LO problem. Each engine can be thought of as a project, and deciding to perform maintenance on an engine is analogous to operating a project. Since there are multiple ways to perform maintenance on an engine, our MAB model will be a superprocess. For engines that are not assigned to maintenance, the state of the engine will deteriorate due to flying activities. This deterioration while performing a resting (i.e., non-maintenance) action means that this model is also restless. The capacity constraints are somewhat analogous to only being allowed to operate m out of the n projects in each period. However, since the model includes multiple resources, there will now be multiple constraints of this type. Finally, the reward gained by an engine is dependent on its state. By providing a reward to only serviceable engines and then maximizing the discounted expected reward over an infinite horizon, we can maximize the number of serviceable engines over an infinite time horizon.

There are two primary differences between the engine maintenance scheduling problem and the typical restless MAB superprocess problem. The first is that maintenance actions can last multiple periods. In the MAB context, this implies an action decision in one period can dictate resource consumption in subsequent periods. Second, the engine maintenance problem involves six different types of resources which requires six different capacity constraints. These resources are the five different module inventories, as well as general maintenance space/ equipment.

3.4.1 State Space

The first challenge in creating our model is to form a tractable state space that captures the relevant information about each engine. In order to make maintenance decisions for an engine it is necessary to know which modules, if any, have unscheduled breakages, the remaining life of each module, and whether the engine is currently in maintenance. We will let the state s of an engine be defined as follows:

- an indicator of which, if any, module is broken. (0-5).

- the amount of life left for module $i \in \{1..5\}$. (0-3).

- an indicator of whether or not engine is undergoing maintenance. (0 or 1)

Therefore, s is a vector containing 7 pieces of information. The first value is an integer ranging from 0 to five and represents which module has experienced an unscheduled break (0 represents no unscheduled break). As already mentioned, although it is sometimes possible for the engine to have more than one module experience an unscheduled maintenance, this is quite rare, and we do not include this possibility in our model.

The second through sixth values are all indicators of the amount of remaining life left in modules 1-5. Ideally we would like this value to be the number of flight hours remaining for the module, since this is the metric by which module life is measured in practice. However, module lives can be thousands of hours long, which would easily make our state space too large to be tractable. Instead, we allow for 3 values of life indicating "high", "moderate", and "low" amounts of remaining lifetime represented by 3,2, and 1, respectively as well as 0 to represent a "dead" module that must be replaced before the engine can be flown.

Finally, the seventh component of our state space is a binary vector, that indicates whether (value 1) or not (value 0) the engine is in maintenance. This state space definition gives us a state space for a single engine with a little over 12,000 possible states, half of which are maintenance states.

3.4.2 Resources and Action Space

Next, we describe the action space for the proposed model. In the traditional restless multi-armed bandit superprocess problem, each bandit can choose to either rest or act in one of a variety of ways. However, in the engine maintenance problem, we have multiple actions which each take varying types and amounts of resources. The first resource we model is maintenance capacity. This aims to model operational capacities, such as manpower, shop space, and the equipment necessary for performing engine maintenance. In the traditional MAB framework, the capacity would be denoted as M. The second type of resource is serviceable module inventory. At each point in time, a maintenance shop might have extra serviceable modules in inventory. As previously mentioned, if modules are used up, new modules must be ordered from the depot, and it can take a few weeks between when they are ordered and when they arrive. We view the inventory of each module as a separate resource. Since there are 5 modules we have 5 different types of respective inventory resources.

Each action in our model takes some combination of resources for some potentially stochastic number of periods. However, the set of allowable actions is dependent on the state of the engine. We, therefore, denote our action space as $U(s)$. We model 3 different types of actions. The first type of action is the "rest" or "passive" action. Rest actions require no resources and is always allowed. However, this action might mean different things for different states. For serviceable engines, the rest action represents all activities that do not include backshop maintenance. In particular, we can assume that engines that are serviceable are in the flying rotation and therefore will degrade over time when taking the rest action. For engines that are broken but not yet in maintenance, the rest action represents waiting for maintenance. The engine will remain broken until a maintenance action is selected. For engines that are in maintenance, the rest action represents remaining in maintenance until all the required work is complete. The rest action is the only action available to engines that are in maintenance.

The second type of action are the replacement actions. In these actions, a subset of the modules of an engine will be replaced. These actions require both maintenance resources and inventory resources, for each type of module that is being replaced. Replacement actions are relatively fast and put engines in good states (all replaced modules have full working life). However, replacement actions also consume modules in inventory, which requires that there exist spare modules in inventory. There are 31 different types of replacement actions as there are 31 combinations of our 5 modules (excluding the empty combination). Note that an engine is in a broken state, some of these actions will be excluded if they do not address all current issues. For instance, if an engine has a broken core module, we do not allow for only a gearbox replacement, as this would not solve the respective engine's problem. The model would, however, allow for a core *and* gearbox replacement. This action type models the "RR" type maintenance described in Section 3.2.1.1.

The third and final type of action is the repair/replace action. In this action type we repair one of the broken modules, and replace any combination of the others. It is important to remember that only unscheduled breaks can be repaired. Modules that have simply run out of lifetime cannot be repaired at the backshop and must be replaced. Repair/ replace actions take longer than simple replace actions, because it is more difficult to fix a broken module than it is to simply replace it.

However, repair/ replace actions do not consume inventory resources of the module type that is being repaired. In other words, if a module is chosen to undergo a repair action rather than a replace action, it takes more time, but no inventory is used. However, repair/ replace actions take more backshop resources since they take more time to complete. There are 80 different repair/ replace actions, but generally most of these are unavailable. These actions are available only when one of the modules has an unscheduled break, and only the actions that include repairs on the right module are eligible to be performed. This action type models the "RRR" maintenance described in Section 3.2.1.1 .

These three action types lead to 112 unique maintenance actions. However, in each specific state, typically only a small subset of these are allowed to be performed. For engines that are serviceable, the model always has the option of replacing any combination of the five modules which gives us 31 possible decisions (plus the rest decision). Each of these decisions will consume maintenance resources, as well as module inventory. The model does not allow for fixing actions on serviceable engines, since there would be nothing to fix. For broken engines, all actions that rectify all current problems are allowed. For instance, if there is an unscheduled breakage, all actions that fix this breakage are allowed. The number of allowable actions for an engine in a broken state varies, depending on the number of unserviceable modules, as well as the reason why these modules are unserviceable.

Finally, the model does not allow for preemption, meaning that once an engine has begun maintenance it continues to take up resources in the maintenance bay until maintenance is complete.

3.4.3 State Transition Probabilities

The state transition probabilities used in this model are both state and action dependent. Since the state space greatly simplifies the relevant information associated with each engine, the resulting transition dynamics are simple, yet reflect the dynamics of the problem as accurately as possible. All of the parameters that are used to describe the transition dynamics are based roughly on historical Air Force data obtained through reported F100-229 breakages over the past 7 years, and through qualitative data obtained in conversation with Air Force practitioners. We choose all parameters in order to emulate one of the operational bases shown in Figure 3.2.1.

First, all engines in serviceable states degrade during each period where the rest action is selected.

This reflects the assumption that all serviceable engines that are not in maintenance are involved in flying operations. Recall that the model does not fully track the remaining lifetime of each module in terms of flying hours, but keeps track of general levels. Thus, during each time period in which a serviceable engine rests, its remaining lifetime transitions to a lower level with probability p and remains the same with probability $(1-p)$. This means that the module lifetime goes from "high" to "medium" , "medium" to "low", or "low" to "expired". The value of p is chosen so that the modules will transition from "high" lifetime to expired in an average number of time periods that reflects reality, on expectation. Specifically, the number of time periods it takes fora new module to expire in the model is distributed according to $NB(3,p)$.

In addition to module lifetimes, the model considers unscheduled breakages. Each time an engine flies it also has a chance of experiencing a unscheduled break. We define the vector q such that $q = (q_0, q_1, q_2, q_3, q_4, q_5)$ where q is the joint distribution describing unscheduled breaks, meaning $\sum_{i=0}^{5} q_i = 1$. Each time a serviceable engine rests, module i breaks with probability q_i. We choose q_i to reflect the average frequency of breakage of module i. With probability q_0 no modules experience unscheduled breaks.

When an engine enters a broken state, its transition dynamics change. Since the engine is broken, we can assume that it is no longer undergoing flying operations. Therefore, a broken engine that takes the rest action experiences a self transition with probability 1.

When an engine is in maintenance, the only available action is to "rest" and remain in maintenance until all necessary work is complete. However, the probability of completing maintenance is different depending on what type of maintenance action is being performed. We assume that all repair actions take longer than replacement actions. In other words, if a decisions is made to repair a module, the number and type of other modules that are replaced does not affect how long maintenance takes. Therefore, if an engine is undergoing maintenance that includes fixing module i, the maintenance action ends with probability h_i and continues for another period with probability $(1 - h_i)$. If an engine is only undergoing replacement actions, we say $i = 0$.

Finally, when an engine is put in maintenance for a replacement action, all modules being replaced return to "high" life and the engine enters a maintenance state. Under a repair action, the module being repaired experiences no change in remaining lifetime, and the engine enters a maintenance state. Table 3.1 gives the transition probabilities that are used. As mentioned above,

Parameter	Value
p	0.0024
q_0	0.97
q_1	0.006
q_2	0.006
q_3	0.006
q_4	0.006
q_5	0.006
h_0	0.1
h_1	0.07
h_2	0.05
h_3	0.05
h_4	0.06
h_5	0.056

Table 3.1: Transition Probabilities

these values are estimations based roughly on historical data.

3.4.4 Objective Function

As stated in previous sections, one of the primary objectives of an engine maintenance unit is to meet the prescribed Air Force WRE goal. Accordingly, there must be sufficient serviceable engines to support all of the active aircraft plus however many WRE engines the base is required to maintain. Generally, maintainers aim to keep as many engines serviceable as possible, while remaining robust to spikes of high demand so that they do not ever fall below the WRE threshold.

In this model, the objective is to maximize the number of engines that are serviceable over the time horizon. To accomplish this, we will use a simple reward function that gives a reward of 1 for every engine that is in a serviceable state and a reward of 0 for all other states. We let $g(s)$ denote the reward gained from being in state s. We define $g(s)$ as follows:

$$g(s) = \begin{cases} 1 & \forall s \in Serv \\ 0 & Otherwise \end{cases}$$

Note that we provide a reward based solely on the state of the engine and not the action chosen.

3.4.5 Linear Programming Formulation

To model the engine maintenance scheduling problem, we make a few modifications to the restless MAB superprocess formulation considered by Bertsimas and Niño-Mora [6]. Our approach is similar to the one detailed in Chapter 2. Since we assume all our engines are identical, we have homogeneous bandits, meaning we consider all engines to have the same evolution process which leads to a reduction in the number of variables in the formulation. In particular, we no longer need a variable for each bandit, but instead have our x variables representing the fleet rather than a single bandit. Second, we introduce more types of resources. Depending on which type of maintenance is chosen, the combination of resources consumed will be different. In particular we add five module inventory resources to our standard capacity resource, giving us a total of six resource constraints. These constraints ensure that on average we do not replace modules more frequently then they are available or place more engines into maintenance than can be accommodated. However, we note that the module inventory constraints are slightly different from the maintenance capacity constraint. In particular, we measure the use of capacity by summing up the proportion of the fleet that is in a state that requires maintenance resources, regardless of which action is being taken. Since the state space includes no information regarding when a module was replaced, we instead measure the use of inventory by summing the proportion of the fleet that performed an action which uses inventory, across all states. With these modifications in mind, the linear program that is used is:

$$\max \sum_{s\in S}\sum_{u\in U(s)} g(s)\cdot x_s^u, \tag{3.1}$$

subject to

$$\sum_{u\in U(s)} x_s^u = \overline{x_s} + \sum_{s'\in S}\sum_{u\in U(s')} \beta\cdot p_u(s's)\cdot x_{s'}^u, \qquad \forall\, s\in S, \tag{3.2}$$

$$\sum_{s\in S_m}\sum_{u\in U(s)} x_s^u \le \frac{\mathcal{M}}{1-\beta}. \tag{3.3}$$

$$\sum_{s\in S}\sum_{u\in Replace_i(s)} x_s^u \le \frac{\mathcal{R}_i}{1-\beta}. \qquad \forall\, i\in[1..5] \tag{3.4}$$

$$x_s^u \ge 0 \tag{3.5}$$

The variables x_s^u represent the total expected discounted amount of time the fleet will spend in state s and take action u . This could also be expressed as:

$$x_s^u = \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{a=1}^{A}\frac{I_{s,u}^a(t)\cdot\beta^t}{A}\right]$$

where A is the total number of engines and

$$I_{s,u}^a(t) = \begin{cases} 1 & \text{if aircraft } a \text{ is in state } s \text{ and taking action } u \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

The objective function maximizes the expected reward while Constraints (3.2) represent the transition dynamics of the problem. In these constraints, $\overline{x_s}$ represents the initial conditions which can either be the fraction of engines in state s at the beginning of the planning horizon, or some distribution over the states. We elect to choose a uniform distribution for $\overline{x_s}$ over all serviceable states, which will give us an offline policy that can be easily characterized. Constraint (3.3) ensures that the total expected discounted fraction of engines in maintenance does not exceed the capacity where $\mathcal{M}$ is the repair capacity divided by the total number of engines and S_m is the set of all maintenance states. We use $\mathcal{M} = \frac{9}{65}$ to represent nine repair slots with 65 engines in the fleet.

Next, Constraint (3.4) ensures that we do not replace modules more often then they are available. In particular, we ensure that the total expected discounted fraction of the fleet that replaces module i is less than $\mathcal{R}_i$, which can be calculated using Little's law. Little's law implies that the average number of modules in the pipeline is equal to the rate at which modules enter the pipeline multiplied by the expected time the modules spend in the pipeline. Since the maximum number of modules that can be in the pipeline at once is the basestock level b_i and the average amount of time spent in the pipeline is v_i, this implies $b_i \geq R_i \cdot v_i$ or $R_i \leq \frac{b_i}{v_i}$ where R_i is the rate at which modules enter the pipeline, which is equivalent to the rate at which modules are replaced. We must then adjust this value by dividing it by the size of the fleet since our variables represent the total expected discounted amount of time the *fleet* spends in a certain state while taking a certain action. Therefore, we obtain $\mathcal{R}_i \leq \frac{b_i}{A*v_i}$. For simplicities sake, we assign $\mathcal{R}_i$ such that $\mathcal{R}_i = \frac{b_i}{A*v_i}$. In particular, we set $\mathcal{R}_i = \frac{b_i}{A*v_i} = \frac{1}{65*10}$ to represent a base stock level of one, an average lead time of 10 days, and a fleet of 65 engines. The set $Replace_i(s)$ is the set of all actions available from state s that involve the replacement of module i.

Finally, β is the discount factor which we set to 0.99. Due to long modules lifetimes, we must "look" sufficiently far into the future to capture the long term effects of our decisions.

3.4.6 Computing Indices

Once we solve this LP , we can use the resultant reduced costs to create indices. In particular, we elect to generate an index for every allowable state/ action pair. We could accomplish this by calculating:

$$\lambda_{s,u} = \gamma_s^u - \gamma_s^{Rest}$$

where γ_s^u is the reduced cost for x_s^u.

One potential problem with this method, however, is degeneracy in the LP. Generally, degeneracy can cause an LP to be difficult to solve and take longer than usual, but will not have adverse effects on the solution once it is obtained. However, in our case, since we directly use the reduced costs generated from our optimal solution, degeneracy can make our index policies work poorly if we do not account for it. Specifically, we can have cases when multiple actions have the same reduced cost of 0, even though only one of those actions corresponds to a basic variable. This would cause all of these actions to have the same index, and would prevent us from being able to intelligently decide between them. To address this, we opt to incorporate the basic variables of our solution into our index policy. In particular, we provide all actions corresponding to primal variables of the LP a small boost in index. We, therefore, opt to calculate our index in the following way:

$$\lambda_{s,u} = \gamma_s^u - \gamma_s^{Rest} + z_s^u$$

where

$$z_s^u = \begin{cases} 1 & \text{if } x_s^u > 0 \\ 0 & \text{otherwise} \end{cases}$$

This allows our index policy to differentiate which actions were selected as primal variables in the case of bad degeneracy.

Heuristic 1 Sign Based Index Heuristic

1. For each engine, find the highest positive indices that are associated with a maintenance action. If an engine has no positive indices associated with a maintenance action, disregard this engine.
2. Remove all indices associated with actions for which the necessary resources do not exist.
3. If no more positive indices exist, stop. Else, choose the highest of these indices.
4. Put the corresponding engine into maintenance for the action suggested by the index.
5. Adjust inventory levels accordingly and go to step 2.

3.5 Index Policies

By applying the techniques outlined above, we obtain a list of indices for each action state pair. However, it is not completely clear how the indices should be used to actually make decisions. Traditionally, if there are m resources available the best course of action would be to select the m engines with the highest indices. However, that may not be possible in this case, since the model used includes multiple resources. In some cases there may exist broken engines with high indices that cannot be maintained due to lack of inventory resources, even though there is maintenance space available. Even if it is possible to put m engines into maintenance, it is not necessarily beneficial to do so. Consider an engine with modules that have moderate amounts of remaining lifetime, none of which are broken. The very small benefit that might be gained by putting this aircraft into maintenance (if any) would be insignificant compared to the loss of the reward provided by a serviceable engine.

To address these issues two heuristic methods are proposed which use the developed indices to make decisions. Both methods operate by looking only at the actions for which the necessary resources are available. The first heuristic ranks each engine according to its highest *positive* index. If an engine does not have a positive index associated with a maintenance action, it is not considered for maintenance. This heuristic places the engine with the highest positive index into maintenance for the action associated with the highest index. Resources levels are then adjusted (number of remaining bays and inventory levels), and the process is repeated until there are either no more resources, or no more engines with positive indices. The heuristic is outlined below:

The second heuristic we introduce is similar, but does not require indices to be positive. Instead it

Heuristic 2 Functionality Based index Heuristic

1. Consider all non-resting indices associated with the broken engines.
2. Remove all indices associated with actions for which the necessary resources do not exist.
3. If no more positive indices exist, stop. Else, choose the highest of these indices.
4. Put the corresponding engine into maintenance for the action suggested by the index.
5. Adjust inventory levels accordingly and go to step 2.

operates under the assumption that, as a general rule, broken engines should be put into maintenance and serviceable engines should not. Therefore, this heuristic considers only broken engines when making maintenance decisions. The heuristic can be described as follows:

Clearly heuristics 1 and 2 are very similar, but differ in the area of which engines are considered for maintenance. Heuristic 1 considers all engines but considers only positive indices, whereas heuristic 2 considers only broken engines but is not concerned with index sign.

One benefit to using offline index policies is that we can easily characterize our policies in terms of several simple operational rules. The general characterization of the indices developed above is intuitive and simple. First, the indices indicate that only broken engines should be maintained. This is most likely due to the objective function that is used, which seeks to maximize the number of working engines that are not in maintenance. If a working engine is placed in maintenance, potential reward is lost in the short term.

When choosing between engines to maintain, the indices prioritize engines that have a long remaining lifetime, and have only a few broken components. For instance, fixing the broken module of an engine that has modules with lots of remaining lifetime is much preferred to fixing an engine that has modules that are all almost dead. This makes sense intuitively, because the same amount of resources are used in both cases, but the former provides an engine that will likely last for a long time, whereas the latter produces an engine that will probably die again soon. This also applies to fixing engines that are moderately unserviceable, as opposed to engines that are severely unserviceable. In particular, the policy tends to assign a low rank to engines that need multiple modules replaced. This is likely because an engine in this state consumes a large amount of resources to become serviceable again, and receives the same amount of reward (the reward associated with one additional serviceable engine). The indices opt to use the available modules to fix multiple other engines, as opposed to only one.

Finally, when given the option between replacing and repairing an engine, the indices generally elects to fix modules with high levels of remaining lifetime, but replace modules with low or no remaining lifetime. However, this part of the policy is highly dependent on the base stock levels and average lead times that we use in our LP model. These values come into play on the right-hand side of the constraints (3.4) from Section 3.4.5. Small changes in these values can cause changes in what modules we opt to repair vs. replace (except for dead modules which we must replace).

3.6 Simulation and Results

To establish the efficacy of the proposed index heuristics, they are tested in an environment that emulates reality . In this section, we present a simulation to obtain an approximation for how the proposed indices might perform in practice under both proposed heuristics methods of applying them. The simulation is based on descriptions of operations from an Air Force F-15E flying unit. Each trial has a planning horizon of 2000 days. On each day 20 of the 25 simulated aircraft in the fleet are required to fly. Since the F-15 is a dual engine aircraft, this means that 40 of the 65 total engines assigned to our base are required to fly each day. A WRE requirement of 10 engines is used, meaning that there must be 50 serviceable engines to satisfy both our sortie requirements and our WRE requirement. This implies that if more than 15 engines are unserviceable on a given day, the WRE requirement cannot be met for that day.

Each day, the engines that are selected to fly a sortie of one to two hours, and are exposed to the chance of experiencing an unscheduled breakage. In addition, the number of flying hours remaining in each module are closely tracked. When an engine flies, the length of the sortie is deducted from the remaining lifetime of each of the modules associated with that engine. Engines that do not fly are assumed to have no change in remaining module lifetime or serviceable status (i.e. an engine will not break when not being flown). We assume a maintenance capacity of 8 engines, meaning that no more than 8 engines can be maintained at once. We also assume that each of the module inventories operate according to a base stock policy with a base stock level of one and stochastic, uniformly distributed lead times between 5 and 15 days.

In each trial of the the simulation, we specify a maintenance policy that is used to determine, in each period, which engines we will maintain, and what maintenance actions will be performed on them. After deciding which engines to maintain, engines are randomly selected from the remaining serviceable engines to undergo flying operations. We simulate the two heuristic methods of utilizing our index policy, as well as a *naive* policy. The naive policy chooses to fix only broken engines. If there are more broken engines then available maintenance slots, the naive policy will choose randomly among the broken engines for which the necessary inventory is available (e.g., if an engine has an expired core module but there is no core module on hand, this engine would not be eligible for maintenance). Once an engine is selected for maintenance, any broken modules will be fixed and any expired modules will be replaced. The naive policy does not replace modules before they expire. In short, the naive policy is a simple policy that does not do anything particularly clever or anything particularly wasteful.

3.6.1 Performance

The results from the simulation detailed above are shown in Figures 3.6.1, 3.6.2, and 3.6.3 and Table 3.2. We are primarily interested in the number of unserviceable engines across the time horizon, and the fraction of days in which the WRE requirement is unmet.

The blue line in each of the figures represents the total number of unserviceable engines for each day and the green line represents the number of broken engines awaiting maintenance. The red line shows the average number of unserviceable engines from the beginning of the time horizon to the current time period.

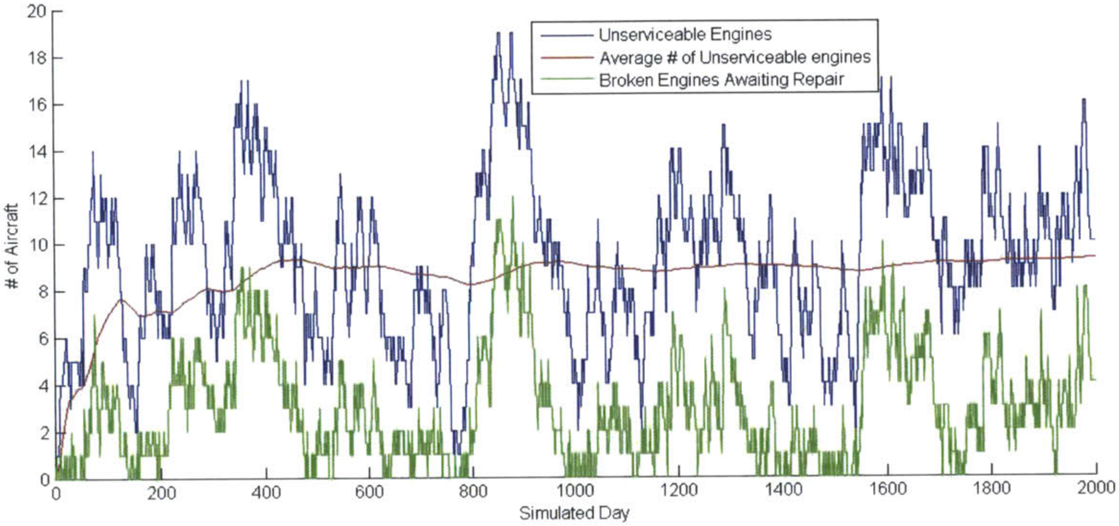
There are a few obvious trends apparent in these figures. First, it is clear that the naive policy performs the worst, giving an average of 9.25 unserviceable engines. In addition, this policy allows more than 15 engines to become unserviceable on a substantial number of days, meaning that the WRE requirement would not have been fulfilled multiple times.

Index heuristic #1 performs slightly better than the naive policy in that it has, on average, only 8.88 unserviceable engines over the time horizon. However, this policy still allows the number of unserviceable engines to rise above 15 frequently, meaning the WRE requirement would, again, not be met in multiple time periods. In addition, index heuristic #1 allows the maximum number of unserviceable engines to rise as high as 20, which is even more than the naive policy, which had a

Table 3.2: Simulation Performance of various maintenance policies

Policy	% of days WRE requirement unmet	Average # of Unserviceable	Max # of Unserviceable over 2000 day planning horizon
Naive Policy	3.75%	9.25	19
Index Heuristic #1	2.5%	8.88	20
Index Heuristic #2	0.05%	7.52	16

Figure 3.6.1: Naive Policy Simulation Results



maximum of 19.

Finally, index heuristic #2 performs better than both the naive policy and index heuristic #1. It gives an average of 7.52 unserviceable engines over the time horizon. This implies that using this heuristic as opposed to the naive policy would allow , on average, 2 more engines to stay in the serviceable stock using the same amount of resources. In addition, this policy only allows the total number of unserviceable engines to exceed 15 one time in the whole 2000 day time horizon.

3.7 Summary

In this chapter we show the power of index policies to aid in the engine maintenance decision process. Although the engine maintenance problem does not match the traditional MAB problem in some ways, the proposed model generates indices that, when used correctly, can lower average unserviceable levels, decrease instances of unmet WRE levels, and shrink variability. We show that

Figure 3.6.2: Index Policy Heuristic #1 Simulation Results

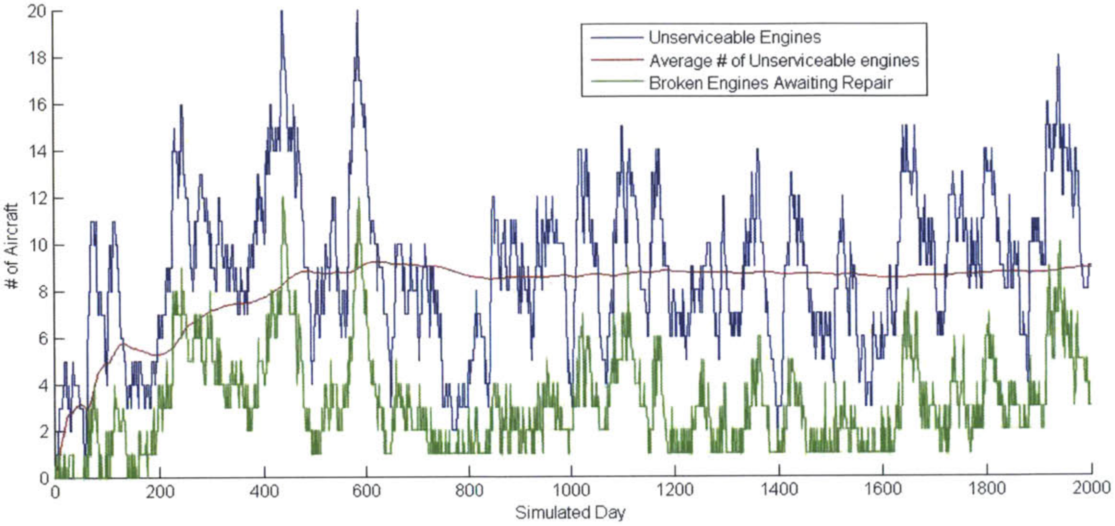
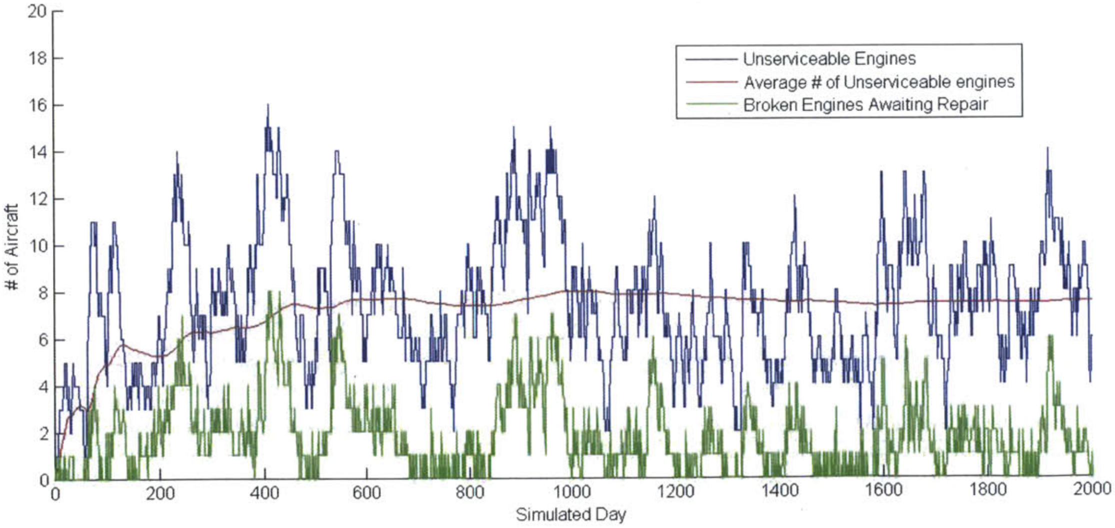


Figure 3.6.3: Index Policy Heuristic #2 Simulation Results



even in a system that utilizes multiple different types of resources and is not always best served by using all available maintenance resources, index polices are still an effective, intuitive, and easily implementable means of decision making.

Historically, the engine maintenance backshop that we emulate in our simulation usually has a WRE stock between 9 and 11 engines and fails to meet WRE requirements about 24% of the time. This indicates a much lower variability and a much higher rate of failure to meet WRE requirements than our simulation suggests. These differences are most likely due to a few factors that were not included in the simulation due to time constraints. Probably the most significant of these is that in reality, the capacity of the backshop is variable. If a backshop fails to meet its WRE requirement, maintenance personnel can be called in on their days off, or manning hours can be increased until the backshop has a big enough WRE stock to go back to normal manning. The network manager can also help to offload work or borrow resources from another backshop. The variable nature of the backshop is not included in our simulation, which causes our variability to increase. The simulated availability of maintenance resources must then be increased so that spikes in demand do not cause the WRE level to hit 0 and adversely affect flying operations (this is something that essentially never happens in reality).

Although the proposed index heuristics outperform naive policies, they do not yet outperform simulated versions of the policies that are currently implemented in maintenance backshops across the Air Force. Although the simulated index heuristics appear to do a very good job of choosing which broken aircraft to work on, they never choose to perform any sort of preventative maintenance (maintenance before the engine becomes unserviceable). Currently, most maintenance personnel will consider replacing modules once they have less than 100 hours remaining. Attempting to replace modules before they expire is beneficial because if the space and modules are available, maintenance can be accomplished with only a small waste of remaining lifetime. If the necessary modules or space are not available, the aircraft can continue to fly since it is not actually broken. We believe that the objective function currently implemented in the LP model disincentives working on any engine that is not broken, because immediate reward would be lost. This could potentially be rectified by supplying some small amount of reward to aircraft that enter maintenance but are still capable of performing flying operations.

Areas of future work include an improved objective function, a more detailed simulation, development of an on-line model, and inclusion of the network nature of the engine maintenance system into the model. The first area may allow the generated indices to be more effective in keeping WRE levels above the Air Force required level by providing incentive to preventative maintenance actions (described above). The second area would allow for a more accurate evaluation of how the proposed index heuristics might perform in practice. The third area would allow for more dynamic policies. In particular, if real-time information about the system was provided to the LP model, it is possible that index heuristic #1 would perform better, as it would have current data on the amount of resources available as well as the current state of each engine. This would allow the positive/ negative aspect of this heuristic to function more accurately.

Finally, and perhaps most importantly, expanding this model to incorporate multiple bases would broaden the scope to a level that might provide more insight to Air Force practitioners on how to use their maintenance network more effectively. Currently, workload and resources are only shared when necessity dictates. It is possible that by preemptively performing resource or load sharing we can decrease the variance of the system even further. Conversations with Air Force practitioners indicate that shipping engines between U.S. mainland bases is relatively inexpensive, meaning that more frequent sharing between bases may be a feasible alternative.

Chapter 4

Conclusions

In this thesis we have shown the power of index policies when applied to real world Air Force scheduling issues. In both the LO maintenance and engine maintenance problems we demonstrate that using a restless multi-armed bandit superprocess framework to generate index policies can give us intuitive and easily implementable way to make decisions regarding real world scheduling problems, even when our problems do not fully match the traditional framework. In particular, we incorporate multiple resources into our models which is not something traditionally included in MAB models.

We begin by presenting the LO maintenance problem and some of the associated difficulties and necessary decisions. We present easily implementable, simple index policies that perform well when subjected to rigorous data-driven simulation. In addition, we gain insight into the importance of good maintenance decisions as well as good flying decisions. In particular, we find that by carefully selecting which aircraft fly to fulfill the daily sortie requirements, we can raise fleet FMC rates by as much as 20 percentage points. In practice, it will never be feasible to make all flying decisions based on LO concerns, however the results from this chapter indicate that taking LO concerns into consideration when making sortie decisions can have large implications for LO FMC rates.

We then present the F100-229 engine maintenance process and outline the difficulties and decisions involved. Again, we generate index policies that provide intuitive rules on when and how to maintain engines in order to satisfy Air Force prescribed WRE levels. We present data-driven simulation results that test the performance of the index heuristics against more naive policies. The results demonstrate the effectiveness of the proposed index heuristics under realistic scenarios.

In particular, the results indicate that using the proposed index heuristics instead of naive policies could decrease the average number of unserviceable engines by almost 20%, and lower the proportion of time in which WRE are not met by a factor of 75.

4.1 Future Work

Although our models produce promising results, there are significant opportunities for future work in both the LO maintenance and engine maintenance problems. Additionally, they both could conceivably be expanded to help in other areas.

4.1.1 LO maintenance

Areas for future work for the LO problem include the incorporation of time varying sortie demands, the inclusion of other types of maintenance, and the inclusion of FOM in our LP model. As discussed in Section 2.8, inclusion of these factors could allow our models to be more accurate, and potentially suggest policies that Air Force practitioners do not currently consider. By incorporating of time varying sortie demands, we can explore how different sortie schedules affect the LO status of the fleet. By including other types of maintenance, our models may find new efficient ways that different maintenance shops can cooperate to minimize aircraft down time while maximizing the amount of maintenance performed during that time period. Air Force practitioners have begun to move LO policy in this direction, indicating that further research in this area would be well received. In addition, inclusion of FOM into our models could allow us to explore how different policies concerning FOM affect fleet LO FMC rates. Currently FOM is prioritized over redux actions. However, it may be possible to opportunistically perform redux on aircraft that require FOM, or to delay FOM altogether in order to perform helpful redux actions first.

Finally, if more detailed data were made available our LO model could be easily be improved. If more accurate data were available, our model could be expanded to differentiate between aircraft with different distributions of damages, as opposed to just SAS number and heavy hitter status.

In recent years, the Air Force leadership has begun to realize the importance of LO concerns and have supplied many additional LO maintenance resources to help keep LO FMC rates up. However, with the recent sequester, these resources might not be adequately manned or supplied. We believe the application of the rules outlined in Chapter 2 could help achieve similar FMC rates with lower levels of resource.

4.1.2 Engine Maintenance

Areas of future work on the engine maintenance problem include an improved objective function, a more detailed simulation, development of an on-line model, and inclusion of the network nature of the engine maintenance system into the model. By altering the LP objective function to provide incentive to preventative maintenance actions, it might be possible to increase the effectiveness of the generated indices at keeping the WRE stock above required levels. Next, a more detailed simulation would allow for a more accurate evaluation of how the proposed index heuristics might perform in practice.

Third, problem lends itself to the development of an online model that could be solved in each period using the current engine states and inventory levels. Because multiple resources are included in the model, each of which can be consumed for a different number of periods due to a single action, the allowable actions in each period could be drastically different even if the states of the engines remain the same. Therefore, it is very probable that the optimal action in each period will be dependent on which resources are available, and in what quantity. Although this type of policy may not lend itself as well to an intuitive set of rules, it would still be simple to use, and would likely perform very well.

Finally, and perhaps most importantly, this model could be extended to incorporate multiple bases. Currently, the network nature of the engine maintenance process is used only in emergency situations. By incorporating the inherent network structure of the modular engine maintenance process into our model, we could devise efficient network management policies which would allow the maintenance backshops to freely share resources and workload. Although some issues regarding tractability would need to be resolved, efficient use of the maintenance network could drastically improve system performance and decrease variability.

Appendix A

Early Work

Section 2.6.4 gives the results from the dynamic programming formulation for a smaller version of the LO model than we discuss in the majority this paper. In this appendix we give the details of this Dynamic Program which was developed by Phil Cho and Eric Zarybnisky. We also present a smaller, simpler LP formulation from Cho and Zarybnisky. Both of these formulations were developed before the model presented in Section 2.5. These older formulations have three key differences from the formulation discussed in Section 2.5. These differences are in the state space definition, action space, and objective function.

A.1 Modeling Framework

A.1.1 State Space

The first large difference, between the model outlined in Section 2.5 and the DP formulation is the choice of state space. In particular since the DP is much more computationally intensive, a smaller state space must be used. To accomplish this, the "residuals" component is left out of the state space vector. This decreases the total number of states to about 6000, which is small enough to make the DP tractable. Since the state space no longer gives information on how much *fixable* SAS exists, different downward transition probabilities must be used, which are given in Table A.1. The smaller LP operates using this same state space and transition probabilities.

This means that the state space is now defined as:

Table A.1: SAS Redux Package Expected Percentage Decrease

	MX Duration (days)			
	1	2	3	4
Non-HH	0.1	0.2	0.29	0.37
HH with SAS $\leq$ 175	0.4	0.47	0.53	0.58
HH with SAS $>$ 175	0.6	0.65	0.69	0.72

- the total SAS number of the aircraft (0-300),
- whether or not the aircraft has a HH (0 or 1),
- what maintenance state the aircraft is in (0-5),
- if in maintenance, the number of days remaining in maintenance (0-11).

The next major difference between the model outlined in Section 2.5 and the models presented below is in the action space. In particular, the "rest" and "fly" actions are combined in the two models presented below. Because of this, the transition probabilities are adjusted to reflect the fact that when an aircraft "rests" it has an approximately 40% (since we assume a fleet of 40 aircraft, 16 of which fly on a given day) chance of flying. This means that flying policies are not developed, though the resulting maintenance policies are tested in combination with the flying heuristics discussed in Section 2.7.1.

The allowable actions are now:

- 1 day maintenance
- 2 day maintenance
- 3 day maintenance
- 4 day maintenance
- 11 day maintenance

Again, given a state vector with the four aforementioned components, there are 6611 possible states.

A.1.2 Objective Function

The final difference between the model outlined in Section 2.5 and the models presented below is the objective function. For these models, different objective function is used, which gives a small amount of reward to aircraft that are not in maintenance, but are also not FMC. The initial rationale behind this was that broken aircraft are still useful to a flying squadron. However, since a LO maintenance shop's primary goal is to keep LO FMC rates high, NFMC aircraft do not contribute to their goal. This is why we instead use the objective function from Section 2.5.4 in our main model.

For the DP model, let $g(s,\lambda)$ denote the reward gained from being in state s. $g(s,\lambda)$ is defined as follows:

$$g(s,\lambda) = \begin{cases} 1 & s \in \text{set of states with SAS number} \leq 100 \text{ and not in maintenance} \\ 0.2 & s \in \text{set of states with SAS number} > 100 \text{ and not in maintenance} \\ -\lambda & s \in \text{set of states that represent being in maintenance} \end{cases}$$

Reward is provided based solely on the state of the aircraft and not the action chosen. This is a result of the fact that once an aircraft is entered into maintenance, it will receive $-\lambda$ for each day it is in maintenance. For a given state s and λ, the DP objective function seeks to maximize the expected future reward:

$$J^*(s,\lambda) = \max_{u \in U(s)} \left[g(s,\lambda) + \alpha \sum_{s' \in S} p_u(s,s') J^*(s',\lambda) \right] \qquad \forall s \in S.$$

In the DP formulation, a discount factor of $\alpha = 0.999$ is used. Large increases in SAS number occur with a low probability, but have a significant impact on the objective function. Accordingly, a high α value is used to "look" farther into the future to capture these effects.

Similarly, for the LP model, $g(s)$ denotes the reward gained from being in state s. $g(s)$ is defined as follows:

$$g(s) = \begin{cases} 1 & s \in \text{set of states with SAS number} \leq 100 \text{ and not in maintenance} \\ 0.2 & s \in \text{set of states with SAS number} > 100 \text{ and not in maintenance} \\ 0 & s \in \text{set of states that represent being in maintenance} \end{cases}$$

A.1.3 Dynamic Programming Based Index Policy

To implement Whittle's heuristic for the LO maintenance scheduling problem we consider every non-maintenance state $s \in S_f \subset S$, where S is the set of all states. For each such state, we calculate λ_s and determine the associated maintenance decision. Note that we do not need to calculate λ_s for $s \in S_m = S \setminus S_f$ as the only feasible decision in these maintenance states is to complete the current maintenance action. This is a result of our nonpreemption assumption. To find λ_s for all $s \in S_f$, we repeat following steps:

1. Begin with a fixed λ value.

2. Solve the discounted infinite horizon DP that models the LO maintenance scheduling problem with a relaxed capacity constraint and corresponding Lagrange multiplier λ.

3. Update λ via bisection search and resolve the DP to find the λ value for which $J^*_{u=\text{no MX}}(s, \lambda) = \max_{u \in M} J^*_u(s, \lambda)$, where M is the set of all decisions to enter an aircraft into maintenance.

4. Set $\lambda_s = \lambda$ once $J^*_{u=\text{no MX}}(s, \lambda) = \max_{u \in M} J^*_u(s, \lambda)$ and $u_s = \arg\max_{u \in M} J^*_u(s))$

We begin by presenting the dynamic program that is solved for each state $s \in S_f$ and a fixed λ. We then explain the bisection search procedure that is used to find the λ that results in $J^*_{u=\text{no MX}}(s, \lambda) = \max_{u \in M} J^*_u(s, \lambda)$.

A.1.3.1 Bisection Search for λ_s

For each non-maintenance state $s \in S_f$, we need to determine a value of λ such that $J^*_{u=\text{no MX}}(s, \lambda) = \max_{u \in M} J^*_u(s, \lambda)$. The maintenance decision associate with λ_s is $u_s = \arg\max_{u \in M} J^*_u(s, \lambda_s)$. For a given state $s \in S_f$ the subsidy value λ_s is found using bisection search. After solving the DP for a given value of λ, if $J^*_{u=\text{no MX}}(s, \lambda) \neq \max_{u \in M} J^*_u(s, \lambda)$, the value of λ is updated.

If $J^*_{u=\text{no MX}}(s, \lambda) < \max_{u \in M} J^*_u(s, \lambda)$, it means that the expected future reward for deciding to enter an aircraft in state s into maintenance is higher than the expected future reward of not entering the aircraft into maintenance based on a subsidy value of λ. In this case, the value of λ is too low (i.e., $-\lambda$ is too high) and λ must be increased. Conversely, if $J^*_{u=\text{no MX}}(s, \lambda) > \max_{u \in M} J^*_u(s, \lambda)$, the value of lambda must be increased.

The initial upper and lower bounds for λ_s are unknown a-priori. This is due to the fact that the value of λ_s is a function of the several factors including the reward function, transition probabilities, and discount factor. However, an upper bound, $\overline{\lambda}$, for λ_s can easily be found. If the DP is solved using $\lambda > 0$ and it results in $J^*_{u=\text{no MX}}(s, \lambda) < \max_{u \in M} J^*_u(s, \lambda)$, then set $\lambda = 2 \cdot \lambda$ and resolve the DP. Continue this process until $J^*_{u=\text{no MX}}(s, \lambda) > \max_{u \in M} J^*_u(s, \lambda)$ at which point a suitable $\overline{\lambda}$ has been found. A similar process can be used to find a lower bound for λ_s, denoted by $\underline{\lambda}$.

Of note is the fact that zero is not necessarily a lower bound on λ_s. Consider a nonmaintenance state with a very low SAS number. In this case, the dynamic program will not place an aircraft in maintenance unless there is sufficient incentive to do so. If $\lambda_s = 0$, the immediate reward in the next time period will be zero if the aircraft is placed in maintenance. If the aircraft is not placed in maintenance, with high probability the immediate reward in the next period will be one due to the empirical SAS creep distribution. In addition, the maintenance action will provide little to no benefit as the SAS number was low to begin with. Accordingly, if $\lambda_s < 0$, the immediate reward in maintenance period will be $-\lambda_s > 0$. Intuitively, λ_s represents the amount an aircraft would be willing to "pay" for a day of maintenance which for a high SAS number can be quite high. For a low SAS number, a subsidy must be provided due to a lost day of operational capability.

A.2 Implementation of the Dynamic Program

The dynamic program described in the previous section was implemented in MATLAB R2010b. Based on computational experience, policy iteration is chosen as our solution method. In addition, since the bisection search for the index values can be done independently for each state, the index policy can be generated in parallel. In total, the computational time needed to generate a complete index policy was approximately 36 hours. Such a significant runtime does not hinder the applicability of this method as the index policy needs to be generated only once, offline, and during execution the precomputed index values and their associated maintenance actions are used as part of the ranking scheme described earlier.

Figure A.2.1 shows the index values associated with all non-maintenance states. The two lines differentiate between aircraft with a heavy hitter and aircraft without a heavy hitter. As expected, for a given SAS number, an aircraft with a HH will always be ranked above an aircraft without

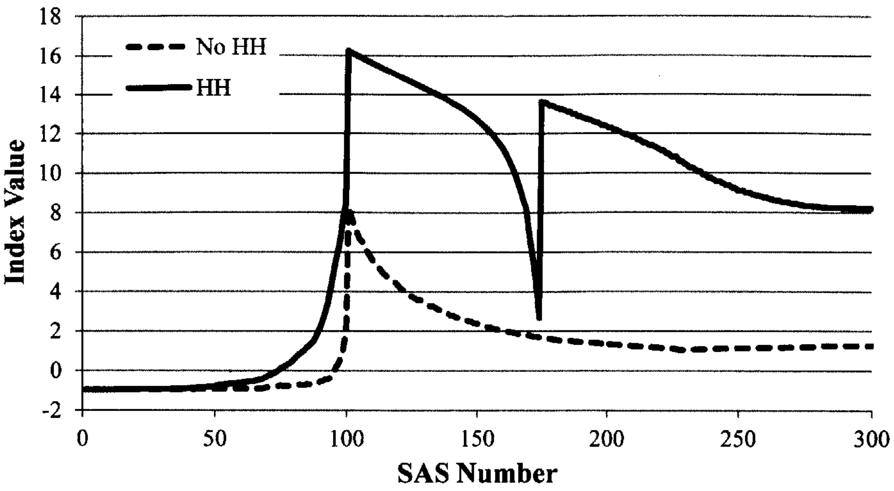


Figure A.2.1: Index values associated with each possible SAS state.

a HH. For both states with HH and those without HH, the index values peak just above a SAS number of 100. This is in line with intuition since aircraft that are slightly above the FMC threshold of 100 are almost guaranteed to become FMC after undergoing SAS redux. The large drop in index values right before states with HHs and a SAS of 175 or higher is a result of the assumption made about canopy damages. Recall that aircraft with a SAS number greater than 175 were considered likely to have a canopy damage and a redux action is likely to eliminate the 150 unit contribution from the single damage.

According to the index values, there are cases when an aircraft with no HHs should be given maintenance priority over aircraft with HHs. For example, an aircraft with no HHs and a SAS number of 101 should be entered into maintenance before an aircraft with a HH and a SAS number of 80. Although this may seem intuitive, historical data from actual flying units show that, often times, aircraft with HH are given priority regardless of other factors. The index values make it possible to easily determine when aircraft with HHs should be given priority and when they should not.

In conjunction to an index value, this procedure yields a state dependent maintenance decision.

A.2.1 Linear Programming Based Index Heuristic

We will also quickly present a more simple version of our LP:

$$\max \quad \sum_{s\in S} \sum_{u\in U(s)} g(s) \cdot x_s^u, \qquad \text{(LPIndex)}$$

subject to

$$\sum_{u\in U(s)} x_s^u = \overline{x_s} + \sum_{s'\in S} \sum_{u\in U(s')} \beta \cdot p_u(s's) \cdot x_{s'}^u, \qquad \forall\, s \in S, \qquad \text{(A.1)}$$

$$\sum_{s\in S_m} \sum_{u\in U(s)} x_s^u \leq \frac{\mathcal{M}}{1-\beta}. \qquad \text{(A.2)}$$

Notice that this is different from the LP presented in Section 2.6. In particular, we no longer have the flying requirement constraint, since we factor flying into our rest action, and use transition probabilities to ensure that enough aircraft fly on expectation. Again, we define γ_s^u to be the reduced cost associated with variable x_s^u. For each non-maintenance state $s \in S_f$ we calculate the index λ_s by:

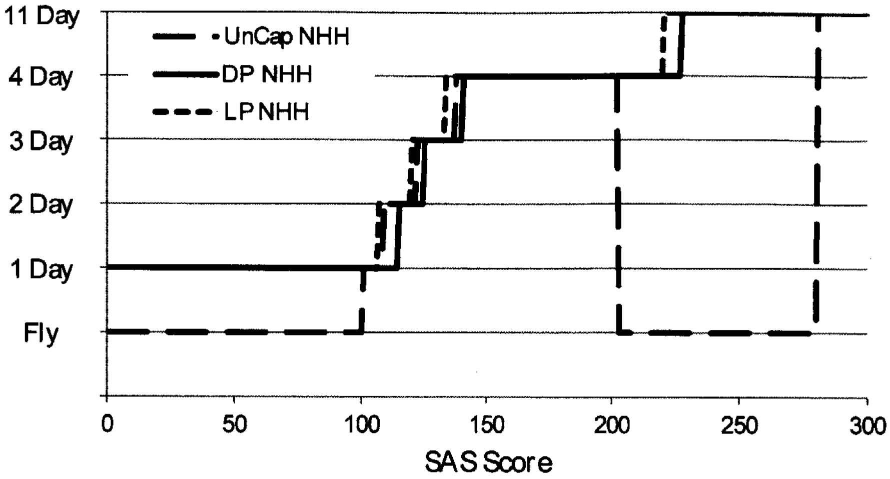
$$\lambda_s = \max_{u\in U(s)\setminus\{noMX\}} \gamma_s^u - \gamma_s^{noMX}. \qquad \text{(A.3)}$$

The maintenance action associated with state s will be the maintenance action that attains the maximum. In contrast to the long running time for the dynamic programming policy, this linear program solves in less than a second. For a number of reasonable starting conditions, the index values generated by the linear program were similar in nature to those shown in Figure A.2.1. There were differences based on the exact starting conditions but large features such as the jump around 100, the canopy damage dip and subsequent jump, and the slow decay of the NHH index were evident in all cases. One particular difference between the DP and LP based indices is a significant rise in the LP index value for high SAS aircraft with no heavy hitters. For all of the starting states tested, the LP based index values for SAS numbers above 280 increased by 60% over the index values for SAS numbers just below 280.

A.2.2 Index Based Maintenance Decisions

Using the index values developed above, maintenance schedulers can easily determine which aircraft should be placed in maintenance once capacity becomes available. In addition to ranking aircraft, these index policies also yield maintenance decisions that determine the SAS redux package an

Figure A.2.2: NHH Maintenance Decisions.



aircraft should be scheduled for.

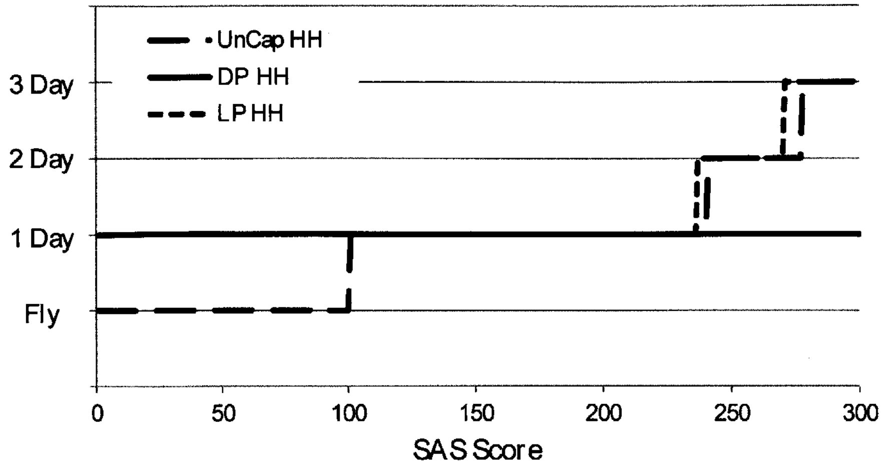
As mentioned previously, the DP based index policy determined the maintenance policy based on the action u_s for which arg $\max_{u \in M} J_u^*(s, \lambda_s)$. As the DP index policy completes an offline calculation, a complete characterization of the maintenance policy can be determined. Figure A.2.2 shows the maintenance decisions associated with each non maintenance state with no heavy hitters. Figure A.2.3 illustrates the corresponding heavy hitter maintenance policy.

Unlike the DP based index policy, the LP based index policy does not generate a maintenance policy a-priori. Rather, the actual state of the fleet must be known and the linear program (LPIndex) solved. The same starting conditions used to characterize the LP based index values were used to characterize the LP maintenance policy. While the maintenance policy varied slightly between different starting conditions, the majority of cases had similar structure as shown in Figures A.2.2 and A.2.3. As a baseline for comparison, the optimal uncapacitated maintenance policy was calculated for each SAS number and heavy hitter state and are shown on the corresponding graphs.

For most HH states, both index policies enter an aircraft into a one day maintenance action. This is expected due to the high marginal returns from the first day of SAS redux on an aircraft with a HH. For large SAS numbers, the LP index policy increases the length of maintenance in a similar manner as the optimal uncapacitated policy. The optimal uncapacitated policy does not perform maintenance on aircraft well below the FMC threshold.

While there is little variation in the HH policies, the NHH policies have significant differences

Figure A.2.3: HH Maintenance Decisions



both across SAS numbers and between policies. In general, as the SAS number increases the associated maintenance action increases in duration. This follows intuition since an aircraft with a higher SAS number needs to undergo a longer SAS redux maintenance action to return to an FMC state. A large deviation between the index policies and the optimal uncapacitated policy occurs over the range [203, 280]. Over this range, the uncapacitated policy does not maintain aircraft but rather continues to fly them. As maintenance capacity do not limit this policy, it simply delays maintenance until aircraft have reached a very high SAS score and then places them in an 11 day SAS redux. In contrast, the both index policies implicitly capture the value of preventative maintenance and place aircraft in this range in 11 day SAS redux.

In summary, each index policy assigns an index value and maintenance decision to each non-maintenance state. Given a fleet of aircraft, each aircraft can be assigned an index value based on the state of the aircraft. After each aircraft has been assigned an index value, the fleet is ordered from highest to lowest using the index values (lowest to highest for LP index). The highest ranking aircraft has the highest priority for maintenance and is entered into SAS redux if there is capacity. When an aircraft is entered into SAS redux, the duration of the SAS redux action is the maintenance decision that is associated with the state dependent index value.

A.3 Simulation

In this section, we give the results of a simulation performed by Cho and Zarybnisky to see how the proposed policies and heuristics might perform in practice. Note that this is a less detailed simulation than the one presented in Section 2.7. The simulation is based on operations data from the same Air Force flying unit that provided the SAS creep data. Each trial has a planning horizon of 1000 days. On each day 16 aircraft are required to fly, 8 from each of two squadrons. Aircraft that fly are subject to SAS increases that follow the empirical distribution presented in Section 2.3. Aircraft that do not fly are assumed to have no change in SAS number. For each trial, we specify the flying rule used to select what aircraft fly in each period. The total fleet size is determined by historical data and is approximately 40 aircraft, although it varies slightly over the course of the planning horizon. Lastly, we assume an LO maintenance capacity of five aircraft.

In each trial of the the simulation, we specify a maintenance policy that is used to determine which aircraft to maintain and what type of maintenance to perform. After deciding which aircraft to maintain, a specified flight determination policy is used to pick the aircraft to fly. We also consider the optimal uncapacitated policy in both an uncapacitated and capacitated context. The optimal uncapacitated policy with unlimited LO maintenance capacity is an upper bound on any maintenance policy. The optimal uncapacitated policy with a limited maintenance capacity serves as a measure of comparison for the index based policies.

Figure A.3.1 shows the average daily FMC rates for 4 simulation runs, each with 1000 periods, with random selection of aircraft for flight. The daily FMC rate is calculated as the number of aircraft not in maintenance with a SAS number of 100 or less divided by the total fleet size. Note that this metric is different than the reward function used in the optimizations. In particular, aircraft with a SAS score greater than 100 contribute nothing to the FMC rate but contribute 0.2 in the optimization framework. This dichotomy is a direct result of the different metrics used at the strategic and operational levels. In most cases, senior leadership is concerned about the FMC rate as it indicates the ability for a flying unit to perform its primary mission, going to war. However, at the operational level, the main objective is to support and sustain ongoing training operations, which may not require a fully stealth aircraft.

In the simulation runs shown in Figure A.3.1, the optimal uncapacitated policy with unlimited maintenance resources far exceeds all of the capacitated policies. Note, however, that the index policies beat the uncapacitated policy by 5-10% when maintenance resources are capacitated. The two index policies have very similar FMC rates with the LP based index policy beating the DP based policy by 1-3%.

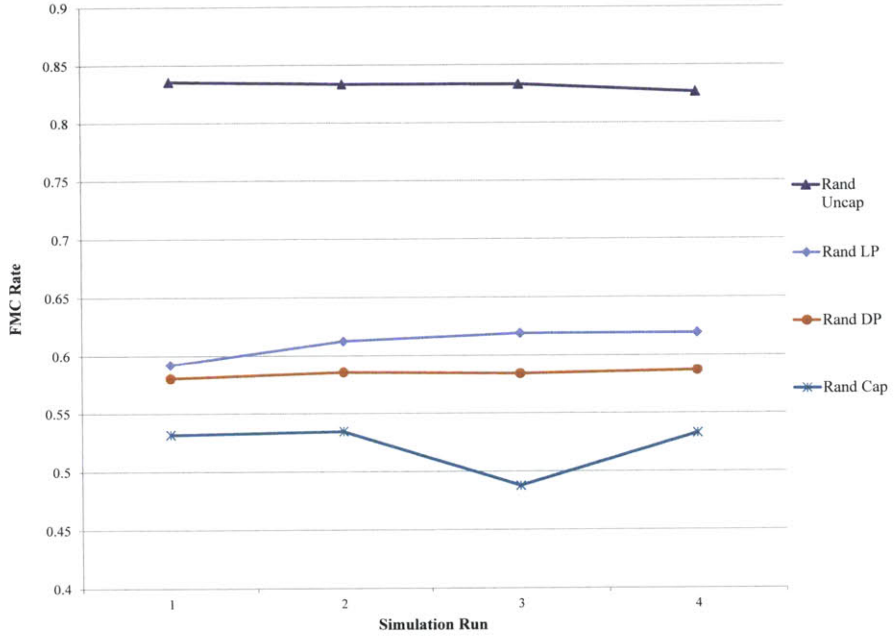


Figure A.3.1: FMC rates for 4, 1000 period, trials with random aircraft selection.

Having established that the optimal uncapacitated policy significantly outperforms the other policies when aircraft are chosen at random for flight, we next consider the other flight scheduling policies discussed in Section 2.7.1. Figure A.3.2 shows the FMC rates for the optimal uncapacitated policy, both with and without maintenance capacity limitations, the DP based index policy, and the LP based index policy for the random, high, and high-low flight assignments. The low selection policy is not shown as it results in significantly worse FMC rates, 20-30% worse than the random selection rule. The FMC rates for the random selection policy in Figure A.3.2 are the same as the FMC rates shown in Figure A.3.1.

The first characteristic of note is the relative indifference of the optimal uncapacitated policy with unlimited maintenance resources to the flight selection rule. Even for the low SAS selection rule (not shown), the average FMC rate for the optimal uncapacitated policy was no less than 79%. This indicates that with sufficient maintenance resources, the selection of which aircraft to fly

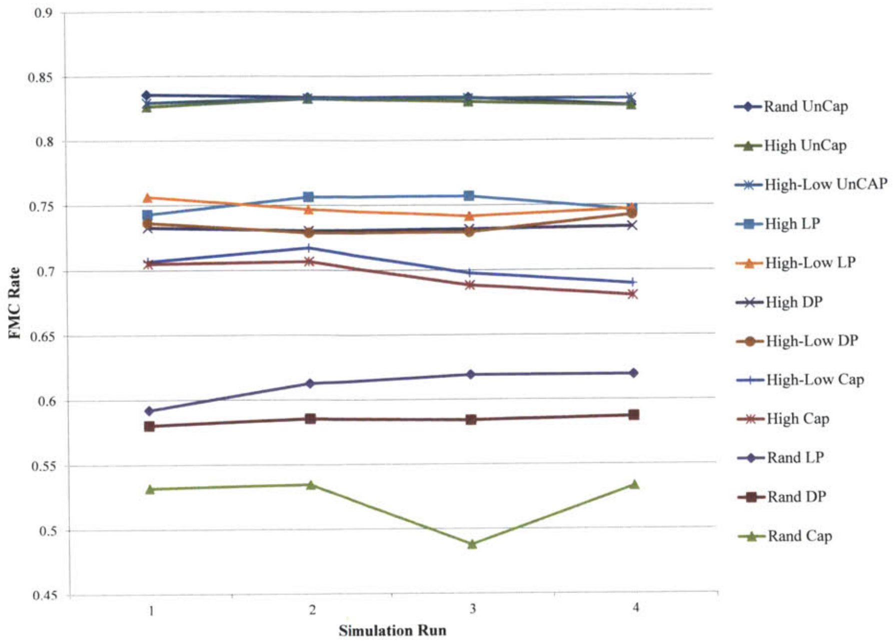


Figure A.3.2: FMC rates for 4, 1000 period, trials for varying maintenance and aircraft flight selection policies.

becomes much less important.

In contrast to the unlimited maintenance capacity case, the choice of which aircraft to fly has a significant impact on the FMC rates for the index and capacitated policies. Interestingly, choosing to fly the aircraft with the highest SAS numbers first results in a significant jump in FMC rate over randomly selecting aircraft to fly. The average FMC rate is approximately 73-76% when employing either the high or high-low selection rules combined with the index based policies. Using the index policies, the random flight selection rule results in FMC rates between 57-63%. This results from the fact that aircraft with low SAS numbers are preserved for the future.

Ultimately, the simulations show the effectiveness of the index policies relative to the optimal uncapacitated policy with limited resources in realistic scenarios. In most cases the LP based index policy outperforms the DP based index policy but by a small margin. In addition, we find that the flying rule used to select the aircraft that will fly in each time period can have a significant impact on the FMC rate.

In support of ongoing discussions within the flying community, further simulations were run to test the effects of varying sortie requirements and maintenance capacity. Simulations were conducted that varied the number of aircraft flown in a day from 10-20 and the LO redux capacity from 4-9 aircraft. The overall fleet size was held constant at 40. While the optimal uncapacitated policy with unlimited maintenance resources is unaffected by the LO redux capacity, the number of aircraft flown had a significant impact on the FMC rate. With a daily sortie rate of 10 aircraft, the FMC rate was approximately 90%. However, the FMC rate dropped linearly as the daily sortie rate increased, with 20 aircraft yielding an FMC rate of 80%. In all cases, the flight selection rule had little to no impact on the FMC rate for the optimal uncapacitated policy with unlimited maintenance resources.

In contrast, those policies limited by maintenance capacity were significantly impacted by the flight selection rule, especially when the LO maintenance capacity was low and the sortie rate high. For instance, with a repair capacity of 4 and a sortie rate of 20, random aircraft selection yielded FMC rates of 17% and 41% for the optimal uncapacitated policy and the DP index policy respectively. However, by using the high-low selection rule, these percentages increased to 52% and 64% respectively.

While the sortie rate had a significant effect on the FMC rate, the LO maintenance capacity had much less of an impact. With a sortie rate of 10 aircraft per day, the increase in LO maintenance capability from 4 to 9 aircraft had virtually no impact (less than 1%), for the high and high-low selection rules. With a sortie rate of 20 aircraft per day, an additional LO maintenance slot increased the FMC rate anywhere between 0.6% (from 4 to 5) and 3.5% (from 7 to 8).

Appendix B

Alternative LP starting States for the LO problem

B.1 Introduction

Recall our LP formulation presented in Section 2.6:

$$\max \sum_{s \in S} \sum_{u \in U(s)} g(s) \cdot x_s^u, \tag{Objective 2}$$

subject to

$$\sum_{u \in U(s)} x_s^u = \overline{x_s} + \sum_{s' \in S} \sum_{u \in U(s')} \beta \cdot p_u(s's) \cdot x_{s'}^u, \qquad \forall\, s \in S, \tag{B.1}$$

$$\sum_{s \in S_m} \sum_{u \in U(s)} x_s^u \leq \frac{\mathcal{M}}{1-\beta}. \tag{B.2}$$

$$\sum_{s \in S_f} \sum_{u \in Flying} x_s^u \geq \frac{\mathcal{F}}{1-\beta}. \tag{B.3}$$

$$x_s^u \geq 0 \tag{B.4}$$

After presenting this formulation we discuss our assignment of the starting state of the LP, $\overline{x_s}$, which comes into play in equation B.1. We decide to use a uniform distribution across all flying states, meaning:

$$\overline{x_s} = \frac{1}{|F|} \qquad \forall s \in F \tag{B.5}$$

$$\overline{x_s} = 0 \qquad \forall s \in M \tag{B.6}$$

Where F is the set of flying states and M is the set of maintenance states. Although this was the distribution we used to generate our final policy, we also explored two other methods of generating values for $\overline{x_s}$. In the first, we take an *online* approach. This means that every simulated day we use the current state of the fleet to assign $\overline{x_s}$, solve the LP, and use the resulting policies to make our flight and maintenance decisions for that day. Our results for this method met with computational issues that ended up giving us worse results than the uniform method of assigning $\overline{x_s}$.

The second method used expected steady state distributions as the $\overline{x_s}$ distribution. The thought behind this was that using the steady state distribution of the fleet across the state space could give the LP more information about how to prioritize different states while still giving us offline policies, meaning our model would only need to be solved once. This method gave no noticeable increase in performance.

B.1.1 Online Approach

As previously stated the online approach for assigning $\overline{x_s}$ involves simulating a fleet of aircraft over time. Each simulated day, the current state of the fleet is recorded, and $\overline{x_s}$ is assigned in the following way

$$\overline{x_s} = \tfrac{Z_s}{A} \qquad \forall s \in S$$

Where Z_s is the number of aircraft in state s and A is the total number of aircraft in the fleet. Since our simulation time horizon is 1000 days, this means we must solve the LP 1000 times. Generally, our LP can be solved within 2 hours using dual simplex, or as little as 10 minutes using barrier methods. Although 10 minutes is relatively short, having to repeat this process 1000 times would make the runtime of the simulation prohibitively long. We therefore search for ways to decrease computational time.

The most effective way to accomplish this was by repeatedly using the dual simplex. Since our LP remains exactly the same except for the $\overline{x_s}$ values on each iteration, and the $\overline{x_s}$ correspond the

Heuristic 3 Steady State Distribution Algorithm

1. Solve LP model using uniform distribution across $\overline{x_s}$, as described in Section 2.6.1. Record index policies.
2. Run simulation using the index policies as decision rules.
3. Record that state of each aircraft over the time horizon, and construct a distribution across the state space.
4. Use this distribution for $\overline{x_s}$ such that $\overline{x_s} = \frac{\sum_{t=1}^{T} Z_{s,t}}{A \cdot T} \quad \forall s \in S$, where $Z_{s,t}$ is the number of aircraft in state s at time t, A is the number of aircraft in the fleet, and T is our planning horizon
5. Re-run LP model using new $\overline{x_s}$ values
6. Return to step 2.

the b vector in standard notation, we can use the optimal basis from the previous day's solution to hot-start the LP. Using this approach, the solve time of each day falls to less than 20 seconds.

Although this method appears promising, and intuitively seems like it would perform better than the method of assigning $\overline{x_s}$ according to a uniform distribution due to the fact that it would supply more current information, it unfortunately does not produce good results. We hypothesize that this is due to computational limits in CPLEX, but the exact reason is not fully understood. The problem seems to stem from premature halting of the dual simplex method. Although the primal variables of the solution are accurate to 10^{-16}, some of the reduced costs of the given solutions are still positive, indicating sub-optimality. Since we generate our indices solely from the reduced costs, this causes our policies to sometimes be very bad. In future work we may be able to use the index boosting method that we introduce in Section 3.4.5.

B.1.2 Steady State distribution

We also try assigning $\overline{x_s}$ according to the simulated steady state of the fleet. We accomplish this using an iterative method that involves both the simulation described in Section 2.7 and the LP model. To accomplish this we use the steps outlines in algorithm 3.

The intuition behind this is that if we can accurately reflect which states are frequently visited and which states are rarely visited in our $\overline{x_s}$ values, this information could help our model create better index policies.

Using the algorithm described above, the $\overline{x_s}$ distributions seem to converge after only 3 or four iterations. However, this method did not appear to help our FMC rates increase. Because of this, we instead use the uniform state distribution, since it is less complex and gives the same results.

Acronyms

FMC	Fully Mission Capable
HH	Heavy Hitter
LO	Low Observable
NFMC	Not Fully Mission Capable
NHH	Non-Heavy Hitter
RR	Remove, Replace
RRR	Remove, Repair, Replace
SAS	Signature Assessment system
WRE	War Ready Engine

Bibliography

[1] Aircraft and Equipment Maintenance Management. *HQ-USAF.* AFI 21-101. 26 July 2010

[2] Aircraft Flying and Maintenance Scheduling Procedures. *ACC Instructions 21-165.* 22 April 2008.

[3] Babaioff, Moshe, Yogeshwer Sharma, Aleksandrs Slivkins. 2008. Characterizing truthful multi-armed bandit mechanisms. CoRR abs/0812.2291.

[4] Bard J., T. Feo "Flight Scheduling and Maintenance Base Planning". *Management Science*, Vol 35, Issue 12, December 1989

[5] Bertsekas D.P. "Dynamic Programming and Optimal Control". No 5 Issue 1. *Athena Scientific optimization and computation series.* Athena Scientific, 2005.

[6] Bertsimas D, J. Niño-Mora. "Restless bandits, linear programming relaxations, and primal-dual index heuristic".1998. *Oper. Res.* 80-90.

[7] Cho, P.Y. (2011) *Optimal Scheduling of Fighter Engine Maintenance*, SM Thesis, Massachusetts Institute of Technology, USA

[8] Cho P., V. Farias J. Kessler, R. Levi, T. Magnanti, E. Zarybnisky. "Maintenance and Flight Scheduling of Low Observable Aircraft" Forthcoming.

[9] Farias V.F., R. Madan. "The Irrevocable Multi-Armed Bandit Problem" Operations Research , Vol. 59, No. 2, 2011, pp. 383-399

[10] Garbe, R. and K. D. Glazebrook. "On a new approach to the analysis of complex multi-armed bandits." *Mathematical Methods of Operations Research* 1998. Vol 48(3).

[11] Gittins, J.C. "Bandit processes and dynamic allocation indices". *Journal of the Royal Statistical Society. Series B (Methodological).* 1979. Vol 41(2) pp. 148-177.

[12] Gittins, J.C., D.M. Jones. "A dynamic allocation index for the sequential design of experiments." University of Cambridge, Department of Engineering. pp 241-266. 1974

[13] Glazebrook, K. and H. Mitchell and P. Ansell. "Index policies of the maintenance of a collection of machines by a set of repairmen". *European Journal of Operations Research.* 2005. Vol 165(1) pp. 267-284.

[14] Gopalan R., K. Talluri "The Aircraft Maintenance Routing Problem". *Operations Research*, Vol 46, Issue 2, April 1989

[15] He, Ting, Shiyao Chen, Hyoil Kim, Lang Tong, Kang-Won Lee. 2012. "Scheduling parallel tasks onto opportunistically available cloud resources". Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing. CLOUD '12, IEEE Computer Society, Washington, DC, USA, 180{187. doi: 10.1109/CLOUD.2012.15. URL http://dx.doi.org/10.1109/CLOUD.2012.15.

[16] Jackson P., and J. Muckstadt "Real-time optimization for aircraft availability". Cornell University Operations Research and Industrial Engineering. Mar 2005.

[17] Kleeman P., G. Lamont "Solving the Aircraft Engine Maintenance Scheduling Problem using a Multi=objective Evolutionary Algorithm" *GECCO '05 Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pp 196-198

[18] Kobbacy, K.A.H., D.N.P. Murthy, R.P. Nicolai, R. Dekker. 2008. Optimal maintenance of multi-component systems: A review. Hoang Pham, ed., Complex System Maintenance Handbook. Springer Series in Reliability Engineering, Springer London, 263-286.

[19] Levi, R., T. Magnanti, J. Muckstadt, D. Segev, and E. Zarybnisky. "Maintenance Scheduling for Modular Systems - New Models and Algorithms", 2011 (Submitted to Naval Research Logistics - Revision Requested)

[20] Muckstadt, J. "A Three Echelon, multi-item model for Recoverable Items". *RAND: Project Air Force.* 1976

[21] Muckstadt, J. A. Analysis and Algorithms for Service Parts Supply Chains. New York, NY: Springer, 2005. Print.

[22] .Papadimitriou, C.H., J.N. Tsitsiklis. 1999. "The complexity of optimal queuing network control". *Math. Oper. Res.* 24(2) 293-305.

[23] Siram C., A. Haghani. "An optimization model for aircraft maintenance scheduling and reassignment." *Transportation Research Part A: Policy and Practice*, Vol 37, Issue 1, pp 29-48, 2003

[24] Tripp, R.S., R. McGarvey, B. Roo, J. Masters, and J. Sollinger."A Repair Network Concept for Air Force Maintenance." *RAND: Project Air Force.* 2013

[25] Varaiya, P., J. and Walrand, C. Buyukkoc. "Extensions of the multiarmed bandit problem: The discounted case". *IEEE Transactions on Automatic Control* 1985. Vol 30(5).

[26] Wang, H., H. Pham. 2006. Reliability and optimal maintenance. Springer Verlag.

[27] Weber, R.R and G. Weiss. "On an index policy for restless bandits". *Journal of Applied Probability* 1990 Vol 27 pp. 637-348.

[28] Whittle P. "Restless Bandits: Activity Allocation in a Changing World." *Journal of Applied Probability*, 25A, pp.287-298, 1988

[29] Whittle, P. "Multi-armed bandits and the Gittins index". *Journal of the Royal Statistical Society. Series B (Methodological)* 1980. pp. 143-149.

[30] Ysebaert S., A. Johnson, J. Miller, T Pettit."An Analytical Approach to Low Observable Maintenance Practices Using Simulation and Marginal Analysis". *IEEE: Proceedings of the 2011 Winter Simulation Conference.* 2011

[31] Zarybnisky, E. (2011) Maintenance scheduling for modular systems-models and algorithms. Ph. D. Thesis, Operations Research Center, Massachusetts Institute of Technology, USA

[32] http://www.f-16.net/f-16_forum_viewtopic-t-6436-postdays-0-postorder-asc-start-45.html

[33] http://www.acc.af.mil/photos/mediagallery.asp?galleryID=611&?id=-1&page=2&count=24