AN APPROXIMATE DYNAMIC PROGRAM FOR ALLOCATING FEDERAL AIR MARSHALS IN NEAR REAL-TIME UNDER UNCERTAINTY

by

Keith W. DeGregory

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Systems Engineering and Operations Research

Committee:

Dr. Rajesh Ganesan, Dissertation Director

Dr. Andrew Loerch, Committee Member

Dr. Lance Sherri, Committee Member

Dr. Alexander Brodsky, Committee Member

signed by Lisa Nolder

Dr. Ariela Sofer, Department Chair

Dr. Kenneth Ball, Dean
Volgenau School of Engineering

Date: 30 April 2014

Spring Semester 2014
George Mason University
Fairfax, VA

An Approximate Dynamic Program for Allocating Federal Air Marshals in Near Real-Time Under Uncertainty

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Keith W. DeGregory
Master of Science
Massachusetts Institute of Technology, 2005
Master of Arts
Webster University, 2002
Bachelor of Science
United States Military Academy, 1997

Director: Rajesh Ganesan, Associate Professor
Department of Systems Engineering and Operations Research

Spring Semester 2014
George Mason University
Fairfax, VA

DEDICATION

This work is dedicated in the memory of all those who have lost their lives to terrorism; to Christine, my beloved wife of 16 years; and to my three beautiful children - Hannah, Daniel, and Benjamin. I pray that my efforts will lead to a safer future so that one day we may all live without fear from random acts of terror.

ACKNOWLEDGEMENTS

This research would not have been possible without the many people in my life who believed in me and provided support in whatever way they could. In no particular order:

Michael Kwinn, thank you for pointing me towards this project. It is only fitting that you had a role in this culminating academic endeavor given that you planted the seed nineteen years ago as my first scholarly role model.

Mark Kukulich and James Curren, thank you for being receptive to and valuing the benefits that academia and applied research can offer to the security of our nation.

To my editors, Valentin Swegle, Jason Southerland, and Christine DeGregory, thank you for devoting time from your busy schedules to carefully scrutinize every sentence, punctuation, and equation in this document.

A heartfelt thank you to my children, Hannah, Daniel, and Benjamin. Your understanding and patience in light of both of your parents concurrently pursuing doctoral degrees has been amazing.

To my wife, Christine, I am at a loss for words – nothing I write here can possibly describe my love for you and appreciation for all you do for our family. You are an extraordinary individual and I am truly blessed to have you in my life.

A loving thank you to my parents, Gilbert and Loretta. You always said you were unsure of where I got my smarts; but one thing is for certain, you taught me dedication and perseverance, without which I would not be writing this acknowledgement.

Thank you to my longtime friend, Josh Helms, for introducing me to the Center for Army Analysis (CAA). Thanks also to Brian Sperling and the rest of the CAA family for opening your doors and providing the resources and full support of a great organization.

Thank you to Stephen Kolitz for setting my post-graduate academic path and research career in the right direction.

Thank you to my committee: Alexander Brodsky, Andrew Loerch, and Lance Sherry for your insightful ideas and tough questioning. You have truly challenged me to produce a product worthy of both the research community and the Federal Air Marshal Service.

Lastly, a special thanks to my committee chair and research director, Rajesh Ganesan. It was an absolute pleasure researching this problem under your guidance and technical supervision. The decision to choose you as my chair has landed me in a *good state* – may following what I have learned through you take me to another *good state* in the future.

TABLE OF CONTENTS

Page

List of Tables ix
List of Figures x
List of Abbreviations xi
List of Symbols xii
Abstract xiii
Chapter One – Introduction 1
  1.1. Research Objective 1
  1.2. FAMS Background 1
  1.3. Quick Reaction Force Strategy 4
  1.4. Methodology 5
  1.5. Contributions 6
  1.6. Structure of Dissertation 7
Chapter Two – Literature Review 8
  2.1. Application Domain 8
  2.2. Commercial Aviation Industry 8
  2.3. Homeland Security and the Federal Air Marshals Service 10
  2.4. Approximate Dynamic Programming 12
Chapter Three – Approximate Dynamic Programming 15
  3.1. Background 15
  3.2. Markov Decision Process 16
  3.3. Dynamic Programming and Bellman's Equation 19
  3.4. Elements of ADP 20
  3.4.1. State Space 20
  3.4.2. Decision Space and Policy 21
  3.4.3. Transition Function 21

3.5. Post-Decision State 24
3.6. Curses of Dimensionality 27
3.7. Value Function Approximation 28
3.7.1. Aggregation 28
3.7.2. Interpolation 29
3.7.3. Function Fitting 29
3.8. Diffusion Wavelets 30
3.9. Phases of ADP Algorithm 33
3.9.1. Exploration 33
3.9.2. Learning Phase and Convergence 33
3.9.3. Implementation (Learnt) Phase 35
Chapter Four – Model Formulation 37
4.1. Overview 37
4.2. Inputs and Parameters 37
4.2.1. Airports and Flights 37
4.2.2. Marshals and Field Offices 40
4.2.3. Critical Time Periods 42
4.2.4. Risks 43
4.2.5. Policy and Decision Space 45
4.2.6. State Space 49
4.2.7. Contributions 50
4.2.8. Working Rules 52
4.3. Model Dynamics 52
4.4. Attribute tracking 55
4.5. Special Considerations 56
4.6. Sample Marshal Routes 57
4.6.1. Two-Leg Work Day 57
4.6.2. Three-Leg Work Day 58
4.6.3. Four-Leg Work Day 59
Chapter Five – Experimentation & Analysis 61
5.1. Experiment Design 61
5.1.1. Controls 61

5.1.1.1. Myopic Strategy ... 61
5.1.1.2. Experienced Scheduler ... 62
5.1.2. Risk ... 64
5.1.3. Metrics ... 64
5.2. Experimental Runs ... 65
5.3. Sensitivity Analysis ... 68
5.3.1. Amount of Risk ... 69
5.3.2. Number of Marshals ... 72
5.4. Observations on Problem Size ... 75
5.4.1. Modeling ... 75
5.4.2. Computation ... 77
Chapter Six – Conclusions ... 80
6.1. Findings ... 80
6.1.1. Methodology ... 80
6.1.2. FAMS Application ... 81
6.2. Broader Impact ... 83
6.3. Future Work ... 84
Appendix A: Airports in Three-Region Model ... 85
Appendix B: Sample Marshal Attribute Tracking ... 86
Appendix C: Sample Allocation Schedule ... 87
Appendix D: Sample Simulation Metrics ... 88
Appendix E: Integer Programming Formulation ... 89
References ... 90

LIST OF TABLES

Table Page
Table 5-1. Problem size comparison.. 76
Table 5-2. State space comparison.. 76
Table 5-3. Computational comparisons of learning phase... 77
Table 5-4. Computational comparisons of implementation phase.................................... 78

LIST OF FIGURES

Figure Page
Figure 3-1: Example Markov chain with 3 states. 16
Figure 3-2: Example Markov decision process with 3 states and 2 decisions. 18
Figure 3-3: State transition model under uncertainty. 22
Figure 3-4: Modified state transition model. 23
Figure 3-5: State transition model using post-decision state. 24
Figure 3-6: Forward dynamic programming algorithm using the post-decision state...... 27
Figure 3-7: ADP algorithm with diffusion wavelet approximation.................................. 32
Figure 3-8: Sample learning rate over duration of simulation... 34
Figure 3-9: Example of convergence to a band 35
Figure 4-1: FAA regions (“Regions and Aeronautical Center Operations,” 2012). 38
Figure 4-2: Critical time periods. 42
Figure 4-3: Example policy for four marshals at time $t = 9$. 48
Figure 4-4: FAMS allocation model 52
Figure 4-5: Two-leg work day 58
Figure 4-6: Three-leg work day 59
Figure 4-7: Four-leg work day 60
Figure 5-1: Mean coverage of high-risk flights (measured against myopic strategy). 66
Figure 5-2: Mean coverage of high-risk flights (measured against experienced scheduler under perfect information). 67
Figure 5-3: Sensitivity on percent of high-risk (measured against myopic strategy). 69
Figure 5-4: Sensitivity on percent of high-risk (measured against experienced scheduler under perfect information). 71
Figure 5-5: Sensitivity on number of marshals (measured against myopic strategy). 73
Figure 5-6: Sensitivity on number of marshals (measured against experienced scheduler under perfect information). 74

LIST OF ABBREVIATIONS

ADP .... Approximate Dynamic Program/Programming
DHS .... Department of Homeland Security
DP .... Dynamic Program/Programming
DTG .... Date Time Group
FAA .... Federal Aviation Administration
FAMS .... Federal Air Marshal Service
IP .... Integer Program/Programing
MDP .... Markov Decision Process
MSE .... Mean Squared Error
OR .... Operations Research/Researcher
PDS .... Post-Decision State
QRF .... Quick Reaction Force
SSI .... Sensitive Security Information
TSA .... Transportation Security Administration
VFA .... Value Function Approximation

LIST OF SYMBOLS

α .... Learning parameter
C .... Contribution function
$d \in \mathcal{D}$ .... Individual marshal decision d in set of all decisions $\mathcal{D}$
$\mathbb{E}$ .... Expectation
$f \in \mathcal{F}$ .... Flight f in set of all flights $\mathcal{F}$
γ .... Discount factor
$m \in \mathcal{M}$ .... Marshal m in set of all marshals $\mathcal{M}$
M .... Transition model
ϕ .... Basis functions
$\pi \in \Pi$ .... Policy π in set of all policies Π
Pr, p .... Probability
$r \in \mathcal{R}$ .... Region r in set of all regions $\mathcal{R}$
$\mathscr{r} \in \mathfrak{R}$ .... Risk type $\mathscr{r}$ in set of all risks $\mathfrak{R}$
s, S .... System state vector
S_t^x, S_n^x .... Post decision State
θ .... Coefficients of basis functions
$\hat{v}$ .... Estimated value of a state
$V, V(S)$ .... Value of being in state S
$\bar{V}, \bar{V}(S)$ .... Approximated value function for state S
w .... Wavelets
W .... Exogenous information random variable
$x = (x_m)_{x_m \in \mathcal{D}, m \in \mathcal{M}}$ ... Decision vector consisting of individual decisions d for marshal m
ω .... Sample realization of exogenous information W
$|$ .... Given
$\in$ .... In
$\sum$ .... Summation

ABSTRACT

AN APPROXIMATE DYNAMIC PROGRAM FOR ALLOCATING FEDERAL AIR MARSHALS IN NEAR REAL-TIME UNDER UNCERTAINTY

Keith W. DeGregory, Ph.D.

George Mason University, 2014

Dissertation Director: Dr. Rajesh Ganesan

The Federal Air Marshal Service provides front-line security in homeland defense by protecting civil aviation from potential terrorist attacks. Unique challenges arise in maximizing effective deployment of a limited number of air marshals to cover the risk posed by potential terrorists on nearly 30,000 daily domestic and international flights. Some risk presents in a stochastic nature (e.g., a last minute ticket sale where suspicion is aroused). Pre-scheduled air marshal deployments cannot respond to risk which presents stochastically in real-time. This dissertation proposes the formation of a quick reaction force to explicitly address stochastic risk of terrorism on commercial flights and presents a method for near real-time force allocation to optimize risk coverage.

The dynamic allocation of reactionary air marshals requires sequential decision making under uncertainty with limited lead time. This dissertation investigates the application of an approximate dynamic program (ADP) to assist schedulers allocating air marshals in near real-time. ADP is a form of reinforced learning that seeks optimal decisions by

incorporating future impacts rather than optimizing only on short-term rewards. The marshal allocation system is modeled as a Markov decision process. Due to the many variables and environment complexity, explicit storage of all states and their values is not possible. Value function approximation schemes are explored to mitigate scalability challenges by alleviating the need for state value storage. The study demonstrates that air marshal allocation in near real-time is possible using an ADP with value function approximation and results in improved coverage of stochastic risk over the myopic approach or pre-scheduling.

CHAPTER ONE – INTRODUCTION

1.1. Research Objective

The research has both theoretical and application objectives. The primary theoretical objective determines the methodology: apply approximate dynamic programming (ADP) to influence decision making under uncertainty in a domain area exhibiting scarce resources and low-frequency stochastic events. As a secondary theoretical objective, the research will also address scalability by demonstrating diffusion wavelet theory as a suitable value function approximator. The application objective is to demonstrate the potential benefits from allocating federal air marshals in near real-time according to optimal policy derived from ADP. The dynamically allocated marshals address stochastic risks and serves to complement the overall security strategy for the Federal Air Marshal Service (FAMS).

1.2. FAMS Background

Federal law enforcement officers first served on commercial airline flights under President John F. Kennedy in 1961 in response to an increase in plane hijackings. The Sky Marshal Program, as it was known then, started with just 18 of these highly specialized officers, but today, the number of air marshals is likely in the thousands[1]. The structure,

[1] The actual number of air marshals currently in service is categorized as Sensitive Security Information (SSI) and is not disclosed in the public domain.

size, and focus of the air marshal program has varied over the decades based on a complex combination of perceived threat, implementation of alternative security procedures, budgetary constraints, political factors, and direct reaction to high-profile incidents. For example, mandatory passenger screening enacted by the Federal Aviation Administration (FAA) in 1973 resulted in the reassignment of nearly all 1,700 air marshals to other positions within the United States Customs Security division. In 1985, following the hijacking of TWA Flight 847[2], Congress rapidly increased the number of air marshals and expanded their mission to focus on U.S. flights operating internationally. Over the years that followed, the number of air marshals continued to fluctuate based on the underlying conditions of the time and the state of the world.

When terrorists hijacked four commercial planes to attack the United States on September 11, 2001[3], the FAMS had only 33 active air marshals in service. President George W. Bush responded to these attacks by ordering the immediate expansion of the FAMS, bringing the number of trained air marshals back into the thousands. Today, the FAMS is part of the Transportation Security Administration (TSA), organized under the Department of Homeland Security (DHS)[4], with the mission to promote "confidence in the nation's civil aviation system through the effective deployment of Federal Air Marshals

---

[2] Members of Hezbollah and Islamic Jihad hijacked Trans World Airlines (TWA) Flight 847, an international flight originating from Cairo destined for London via Athens and Rome, on June 14, 1985. The hijackers held the 153 passengers and crew hostage as a bargaining chip for the release of Shi'ite Muslims in Israeli custody.

[3] During these attacks, 19 al-Qaeda terrorists simultaneously hijacked four commercial airliners in flight with the purpose of using the jets as projectiles to target key U.S. government buildings and infrastructure. The attackers were successful in three of the four attacks with the fourth thwarted by the heroic efforts of the crew and passengers of United Airlines Flight 93.

[4] FAMS was transferred from U.S. Immigration & Customs Enforcement (ICE) to TSA on October 16, 2005 as part of realignment following recommendations and findings stated in DHS Second Stage Review (USGPO, 2007).

(FAMs) to detect, deter, and defeat hostile acts targeting U.S. carriers, airports, passengers, and crews" ("Federal Air Marshals," 2013). Placement of air marshals on commercial flights promotes confidence, but the key to success lies in the "effective" deployment of air marshals. No formal definition for effectiveness of air marshal deployments was found in public domain. For this research, effective deployment is defined as maximum coverage of perceived high-risk flights for a given high-risk threshold setting. The risk threshold setting is discussed in detail in section 4.2.4.

Merely increasing the number of air marshals in force is not sufficient as an effective security strategy. Even at their healthiest numbers over the decades, the air marshal-to-flight ratio on a given day is small. Thus, deployment strategy of this limited resource is critical to carrying out the FAMS mission and minimizing hostile acts via the commercial air industry. Air marshals are the last line of defense against would-be terrorists who have passed through security and screening to board a passenger aircraft. It follows that the FAMS strategy should include analyzing passenger manifests and devoting resources to flights whose manifests pose the greatest uncertainty and thus exhibit higher levels of potential risk. Employing such a flexible strategy for the entire force might be operationally prohibitive for many reasons including low schedule predictability, numerous unplanned nights away from domiciles, and increased expenditures on per diem and other real costs required to support such a strategy. The research proposes the formation of a smaller, more agile reactionary force to mitigate the dynamic and stochastic risk related to the composition of flight manifests. Assuming the FAMS already implements strategies to effectively protect against deterministic sources of risk such as aircraft size and flight paths that threaten critical and vulnerable infrastructure, then the implementation of a Quick Reaction Force (QRF) would complement such a strategy by countering the stochastic sources of risk.

1.3. Quick Reaction Force Strategy

The QRF strategy poses a number of unique and challenging problems with regard to allocation and positioning of scarce resources in order to maximize coverage of high-risk flights. Due to the number of variables, size of the air transportation system, and the stochasticity of the risks involved, the problem is essentially NP-hard and thus, impossible to solve to optimality. Furthermore, in many cases, opportunity to respond to the stochastic risk has very limited lead time, so, the ability to make decisions in near real-time is critical to an effective QRF strategy. The research objective in support of the application is to find an optimal policy for implementation in near real-time. Policy in this context refers to the set of individual decisions for all marshals at a single point in time. Decisions and policy are addressed further in section 4.2.5. The concept of an optimal policy is described below.

Each successive implementation of the policy must account for the combined effects of the individual decisions on all QRF marshals currently available in the system. To incorporate the future impact of decisions, the policy must be able to look past immediate short-term rewards in favor of optimally positioning marshals throughout the system to maximize the opportunity to react to emerging stochastic risks. A simulation-based implementation of ADP produces an optimal policy for allocating marshals by learning the best policies (given the system state) that achieve maximum coverage of high-risk flights over an infinite horizon (large number of iterations in a simulation).

1.4. Methodology

The optimal scheduling of thousands of air marshals on tens of thousands of flights is an intractable problem. The use of heuristics and other innovative approaches can lead to near-optimal schedules in reasonable time. The FAMS has developed procedures and methodology to produce feasible schedules targeting risky flights while adhering to all constraints on work conditions. The exact procedures employed by the FAMS, as well as the data, metrics, and performance of those procedures, are confidential. However, in 2006, the Homeland Security Institute conducted an independent assessment of the FAMS operational approach finding it to be reasonable (Lord, 2010).

Inaccessibility to the data and existing procedures poses a different set of challenges for the researcher. Due to sensitivity of the FAMS operations and procedures, information about the existence and/or extent of a QRF is unavailable. Regardless, merely having a QRF as part of strategy does not automatically equate to improved effectiveness; more important is the means of allocation of the QRF marshals. As a result of this research, the FAMS may consider formation of a QRF if one does not exist or may improve allocation procedures for an already formed QRF.

The QRF primarily provides benefit by mitigating stochastic risks, which in this research represents inherent uncertainty in passenger manifests. As an additional complication, manifests do not become final until hours or minutes before a flight's scheduled departure. Allocating air marshals in response to a positive risk signal will affect distribution of QRF marshals throughout the system at the next decision point, which could merit repositioning other marshals to maintain optimal balance.

The methodology allocates marshals to flights by using an approximate dynamic program, which supports time-constrained sequential decisions under uncertainty. The problem is formulated as a Markov decision process over an infinite time horizon. The process derives optimal policy from unsupervised learning of system state values through simulation. Computational challenges arise from large problem and outcome space, but aggregation and value function approximation mitigate complexity. As a result, the methodology and decision support tool will assist schedulers in determining the optimal allocation of a QRF to cover maximum risk.

1.5. Contributions

The contributions from this body of work touch on both the methodological and applicative domains. Primarily as a contribution to theory, the research demonstrates ADP as an effective approach to developing optimal policy in view of scarce resources and low-frequency stochastic events. Furthermore, the research addresses and tests scalability using diffusion wavelets for approximating the value function used in the ADP. Use of diffusion wavelets as a value function approximator has recently surfaced in the literature and has only been applied as a proof of concept. Successful implementation of diffusion wavelets in this research serves as further evidence of supporting the use of diffusion wavelets as a viable approach to approximating value functions. Better and more efficient approximating functions address scalability concerns and facilitate solving larger and more complex problems.

By proposing a QRF strategy in conjunction with an optimal allocation scheme developed from ADP, the research contributes to more proactive allocation of air marshals in countering the adapting strategies of adversaries to the United States. The policy considers the stochastic nature of the risk and can be implemented in near real-time. Lastly, the security segment of the air transportation industry often receives less attention from the research community. This research extends operations research (OR) support to this component of the industry.

1.6. Structure of Dissertation

Chapter 2 contains a review of the relevant literature in the application domain as well as on the methods employed in the research. In Chapter 3, the essential elements of ADP lay the groundwork for application to the problem space. Chapter 4 describes the model and simulation environment in detail with discussion of the ADP algorithm as incorporated into the simulation. Chapter 5 describes the experimental design, findings from the experiments, and discussion on parameter sensitivity. Chapter 6 summarizes the research conclusions, presents broader impacts of the study, and addresses future work.

CHAPTER TWO – LITERATURE REVIEW

2.1. Application Domain

This problem spans two domains: the commercial aviation industry and homeland security. Both domains have benefited greatly from the study and application of operations research techniques over the last half-century. The wide range of applications and improvements from OR are outlined by Clarke and Smith (2004) for the airline industry and by Wright, Liberatore, and Nydick (2006) for homeland security. The FAMS problem poses unique challenges not found in either of these domains.

2.2. Commercial Aviation Industry

The commercial aviation industry has long recognized the power of OR. By applying OR to industry operations, commercial aviation has vastly reduced the waste and inefficiencies that have contributed to the failures of multibillion dollar airlines. In particular, OR has direct influence upon areas such as resource allocation and utilization, scheduling, capacity management, and flow control. Barnhart, Belobaba, and Odoni (2003) provide an excellent overview in of the many areas of the industry that have employed OR. The commercial aviation domain remains rich with large and interesting problems for operations researchers to study and improve through application of the trade tools. Individual airlines and the FAA often employ operations researchers on staff. A quick web search will reveal hundreds of recent scholarly articles and papers on the subject and many organizations affiliated with the industry.

A popular component of the commercial air industry among researchers is addressing airline crew scheduling due to its many rich features and vast potential for optimizing costs for the airlines. Barnhart et al. (2003) discuss the many variants of the problem, but one common theme is the generation of a subset of feasible pairings for which each flight is included in exactly one pairing. The pairings are selected so that sequence of flight legs in the pairing originate and terminate at the same location and satisfy the work rule constraints for the crews in terms of consecutive work hours. The possible combinations of all pairings for the full scale problem is enormous so only evaluating a small subset of pairings is possible. This problem, at its core, is a set partitioning and set covering problem and is difficult due to the complexity of the system and large number of integer variables. Many solution approaches exist and are discussed by the authors.

This particular problem is of interest to the FAMS allocation problem because under normal operating conditions it is important to minimize the number of nights a marshal spends away from his domicile. However, differences between the problems exist in that air marshals are not required to cover every flight as in the crew scheduling problem, nor could they due to the limited supply of marshals. Regardless, one approach to scheduling air marshals is to generate feasible flight pairings and pre-schedule a marshal's entire work day by selecting pairings that maximizes the coverage of perceived risk. The drawback here is that such an approach is only suitable for deterministic risks and exhibits large processing times ill-suited for allocating marshals in near real-time under uncertainty.

Unlike the commercial aviation industry, the FAMS has received far less attention. One possible explanation for the oversight includes the necessary confidentiality surrounding data and operations. A second explanation might be a perceived absence of a direct linkage to profitability. However, this perception is debunked by Asay and Clemons (2009). The authors developed an analytical prediction model to assess the impact of the 9/11 attacks. They found an immediate 31% to 44% drop in passenger travel following the attacks. Travel rapidly rebounded over the following year, but it ultimately fell short by 3% to 8% of long-term projections calculated in the absence of such events. Proactive risk reduction could indirectly benefit the airlines' bottom line; more importantly, more effective resource application ensures the maximum benefit achieved from the funds appropriated by Congress for law enforcement on commercial air travel.

By applying innovative OR approaches similar to those used in the commercial aviation industry, FAMS is likely to benefit in the same manner with regard to its business and operations.

2.3. Homeland Security and the Federal Air Marshals Service

The FAMS organization has not gone entirely unnoticed by the research community as a small number of papers address specific aspects of FAMS' operations. A value-focused thinking (VFT) approach is used to optimize field office allocations (Castelli, Meier, Morris, Philie, & Kwinn, 2013). The authors analyzed the assignment of marshals to field offices across the country by considering both risk and cost. They applied a two-phased approach of VFT to achieve near-optimal solutions. This research addressed an important facet of the overall strategy in support of the FAMS mission – maintaining optimal numbers of marshals at strategic locations.

Tsai, Kiekintveld, Ordonez, Tambe, and Rathi, (2009) modeled the problem of scheduling marshals as a Stackleberg game and introduced software assistants to employ randomization to air marshal scheduling. The Intelligent Randomization in Scheduling (IRIS) system developed under the research mitigates perception of predictability by always targeting the top tier of riskiest flights. The FAMS employed IRIS as a pilot program for scheduling marshals on international flights operated by U.S. carriers. Due to security considerations, the effectiveness of the pilot program is not accessible in public domain. Jain et al. (2010) elaborates on the IRIS system but also discusses software assistants for randomized patrol planning in the context of airport police at Los Angeles International Airport. While clearly the missions of airport police and air marshals differ, the problems share characteristics with respect to leader-follower dynamic in game theory modeling. Modeling the interaction between the adversary and the air marshal is outside the scope of this research, but it is recognized that incorporating this dynamic into the air marshal allocation model could have added benefits.

The implementation of an effective allocation strategy for QRF marshals would rely on a system that monitors and assesses risk levels of passenger manifests in near real-time. A number of initiatives to achieve this have been proposed since 9/11 but ultimately not implemented over privacy concerns. (Barnett, 2004) summarizes the vision of the CAPPS (Computer-Assisted Passenger Prescreening System) II initiative under TSA. The CAPPS II would make threat assessments of individual passengers from processing thousands of details on individuals, both personal and private. However, a number of organizations such as the American Civil Liberties Union raised concern over privacy and civil liberties issues, and TSA terminated CAPPS II in the midst of controversy. In 2009, the TSA began implementation of a system called Secure Flight, which matches passenger information against federal watch lists. While CAPPS II and Secure Flight share some functionality, the TSA avows not to collect or use commercial data. Secure Flight originated from a recommendation from the 9/11 commission that the TSA should maintain uniform watch lists rather than the airlines. Implementation of Secure Flight completed in November 2010 (“Secure Flight Program,” 2014).

2.4. Approximate Dynamic Programming

ADP spans many applications and problem classes. As researchers continue to develop approaches to mitigate the scalability of large problems, it becomes increasingly popular as a solution approach. ADP is well-suited and designed for solving problems requiring decision making under uncertainty. Simão et al. (2008) apply an approximate dynamic program to large-scale fleet management for Schneider National, one of the largest truckload motor carriers in the U.S. with over 6,000 long-haul drivers. In this work, the authors met their goal of obtaining solutions to extremely high-dimensional state variables that closely matched the performance of a highly skilled group of dispatchers. This application bears resemblance to the marshal allocation problem due to similarities in the physics and dynamics of the system model. The authors use aggregation to approximate the value function, determine an optimal step-size algorithm for the learning rate, and implement a double pass algorithm to speed up convergence. Aggregation ideas from this research serve as a basis for the levels of aggregation studied in the ADP formulation for the marshal allocation problem.

Addressing other aspects of airport security, McLay, Lee, and Jacobson (2010) utilized risk-based policies for checkpoint screening at airports. The authors modeled the sequential passenger assignment problem as a MDP and use ADP to compute the optimal policy of assigning various security devices to passengers assessed at different risk classes upon check-in. It is not clear whether air marshals are considered as a security “device” in the problem. However, seeing how marshals serve as the last line of defense in the airline security system, it follows that their inclusion in the model could serve as an effective cueing mechanism for the employment of QRF air marshals.

McLay et al. (2010) found the optimality equations for the Sequential Stochastic Multilevel Passenger Screening Problem (SSMPSP) to be computationally intractable and can only be solved in a reasonable amount of computation time for small instances of SSMPSP. The authors' research results in a Sequential Stochastic Assignment Heuristic (SSAH) that efficiently approximates the solutions to SSMPSP in real-time. The formulation of the ADP in the FAMS allocation problem draws on techniques employed in the SSAH to facilitate near real-time approximations calculated in the algorithm.

With careful consideration during design, selection of an appropriate approximator for value functions results in implementable approximate dynamic programs. A growing body of work defines the scope and nature of approximation considerations. George, Powell, Kulkarni, and Mahadevan (2008) considered the right level of aggregation and weighting schemes for multi-attribute resources to facilitate value function approximation.

The authors proposed an adaptive weighting strategy that is implementable for very large-scale problems exhibiting large attribute space.

Mahadevan and Maggioni (2006) suggested the use of diffusion wavelets and Laplacian eigenfunctions for value function approximations in solving large scale MDPs. Balakrishna (2009) employed diffusion wavelet-based value functions as a proof of concept to deal with the large dimensionality and significant uncertainty as applied to the problem of taxi-out time estimation. The author also applied ADP to assist in predicting taxi-out times. In an adapted form, her model forms the basis for the system environment for the air marshal problem.

CHAPTER THREE – APPROXIMATE DYNAMIC PROGRAMMING

3.1. Background

Dynamic programming (DP) dates back to the 1950s when Richard Bellman published his "Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" (Bellman, 2003, p. 83). At the same time, other engineering disciplines and research communities were independently converging upon the same discoveries. As a result, dynamic programming is known under different names such as reinforcement learning, neuro-dynamic programming, and adaptive dynamic programming.

Dynamic programming is very efficient at solving problems of small size for which transition probabilities are known. With larger problem spaces, transition probabilities are almost certainly unknown, but they are a necessary component of dynamic programming. The uncertainty inherent in larger problem spaces led to the development of approximate dynamic programming, which avoids the need for transition probabilities through a simulation (learning-based) framework while providing many other added benefits.

Due to its use across many fields, DP/ADP notation can become confusing. As used in this research, notation and definitions are adapted from (Powell, 2011). The description of ADP found in this chapter serves as the conceptual basis for understanding the ADP framework and its application to the FAMS' allocation problem.

3.2. Markov Decision Process

Markov chains and the Markov decision process provide the template for DP/ADP modeling and algorithms. This section provides an overview of the MDP and related research. Consider the directed graph $G = (S, E)$ depicted in Figure 3-1 and described by the countable set of states $= (s_1, s_2, \ldots, s_{|S|})$ and set of edges E.

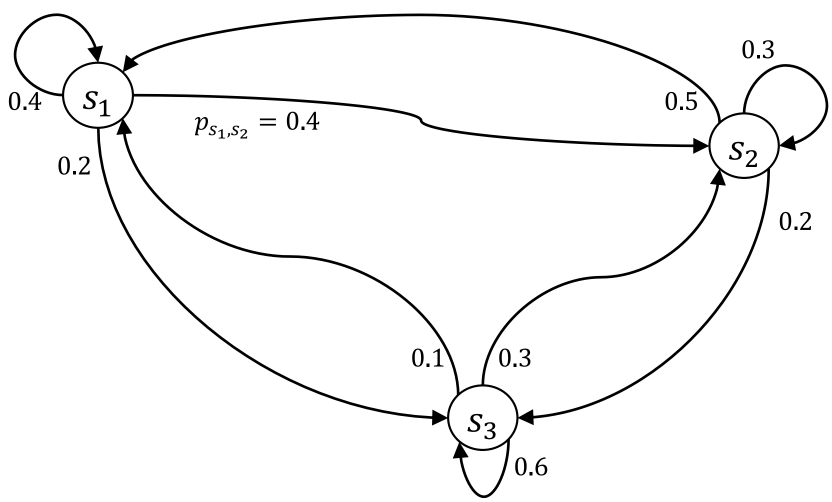


Figure 3-1: Example Markov chain with 3 states.

System transitions occur randomly at uniform time steps from state s_i to state s_j with probability p_{s_i,s_j}. Each transition only depends on the current state s_i, rather than the history of preceding transitions bringing the system to s_i. Such a system is a discrete-time Markov chain and exhibits the Markov memoryless property which states

$$\Pr(S_{t+1} = s_j | S_t = s_i, S_{t-1} = s_k, \dots, S_1 = s_1) = \Pr(S_{t+1} = s_j | S_t = s_i), \qquad i, j, k \in S. \tag{3.1}$$

MDP is a variant of a Markov chain in which the transition between states is partly random and partly controlled by a decision maker taking action x from a set of actions $\mathcal{X}$ at each time step. Decisions result in a reward or contribution C. Thus, the decision influences the probability of transitioning from state s_i to s_j and is given as

$$p_{s_i, s_j}(x_k) = \Pr(S_{t+1} = s_j | S_t = s_i, X_t = x_k) \tag{3.2}$$

with the resulting contribution of $C(S_j | S_i, x_k)$. Figure 3-2 shows the analogous Markov chain from Figure 3-1 but with 2 decisions incorporated into the system transforming the chain into a MDP. The decisions in the MDP are depicted by the gray circles and all transitions from one state to another must pass through a decision point. The decisions and probabilistic branches each have associated rewards and penalties depicted in color next to the transition probabilities.

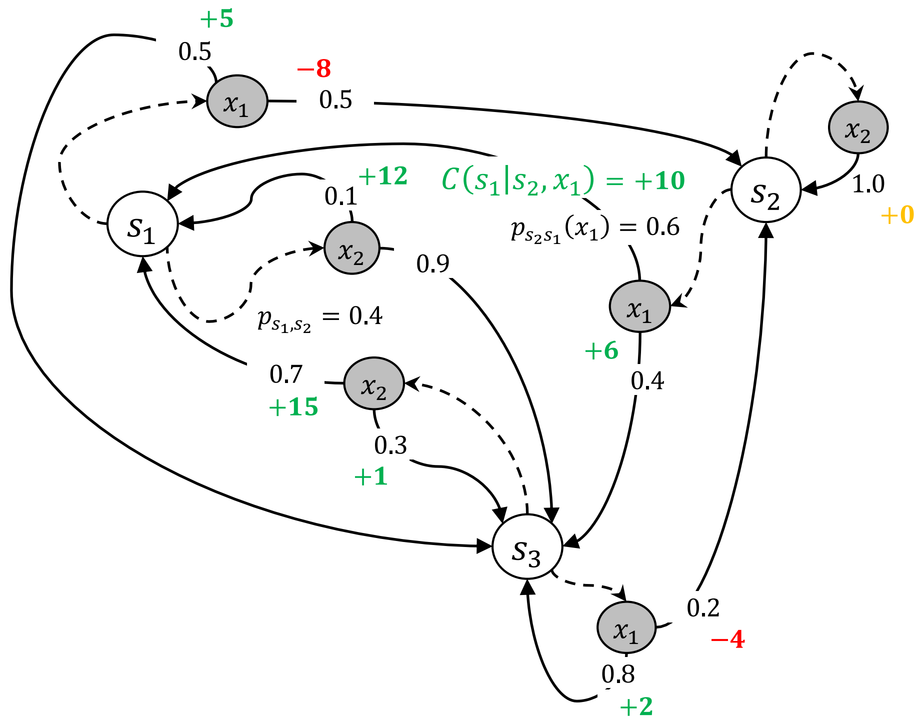


Figure 3-2: Example Markov decision process with 3 states and 2 decisions.

The MDP purpose is to find the best policy $X_t^\pi(S_t)$ to follow to produce the greatest expected contribution over all outcomes. Thus $X_t^\pi(S_t)$ represents a particular policy from the set of all policies Π given the system is in state S_t. The best policy in an MDP maximizes some cumulative objective function of the potentially random contributions over a long run horizon

$$\max_{\pi\in\Pi} \mathbb{E}\left\{\sum_{t=1}^{T} \gamma^t C_t^\pi\left(S_t, X_t^\pi(S_t)\right)\right\}. \tag{3.3}$$

3.3. Dynamic Programming and Bellman's Equation

The entire optimization problem in equation (3.1) does not need to be solved at once. Under dynamic programming the problem is solved iteratively through recursive updating of a value function $V_t(S_t)$ which holds the value of being in the state S_t. Take note that the value of a resultant state one time step in the future is discounted by γ, typically set to a value of 0.9 or 0.95. Discount factor accounts for S_{t+1} occurring one time step in the future. Supposing all state values are stored, then the current value of the new state is represented by $V_t(S_{t+1})$. As a result, the decomposed optimization problem is stated as

$$V_t(S_t) = \max_{x_t \in X_t} (C_t(S_t, x_t) + \gamma \mathbb{E}\{V_t(S_{t+1}) | S_t\}). \tag{3.4}$$

This equation is commonly referred to as Bellman's equation and is written in standard form where the expectation is encapsulated in the max operator. Solving Bellman's equation is easy on small problems where the transition probability matrix is known. However, in larger problems, transition probabilities are often unknown and impossible to calculate. This issue is alleviated through the use of a post-decision state described in section 3.5.

Dynamic programming solves the above optimization problem rooted in the MDP iteratively by breaking the problem into the sum of two components: the immediate contribution $C_t(S_t, x_t)$ from decision x_t taken from state S_t and the discounted expected value of the next system state $V_t(S_{t+1})$. The resultant problem becomes

$$V_t(S_t) = \max_{x_t \in \mathcal{X}_t} \left(C_t(S_t, x_t) + \gamma \mathbb{E}\{V_t(S_{t+1}) | S_t\} \right). \tag{3.5}$$

Simulation iteratively solves this decomposed problem to find the long run value of all states $V(S)$. This learning-based simulation makes it possible to solve large problems that are intractable using classical mathematical programming approaches.

3.4. Elements of ADP

3.4.1. State Space

The state variable S_t captures the information required to describe the system status at time t and holds the information necessary to inform decisions. As defined by Powell, "a **state variable** is the minimally dimensioned function of history that is necessary and sufficient to compute the decision function, the transition function, and the contribution function" (2011, p. 179). The state variable represents a state in a Markov chain. Often expressed as a multi-dimensional vector, the state variable results in an extremely large state space for large problems and restricts the scalability of ADP. In the case of a resource allocation problem, the state variable might include available resources R_t and demands D_t waiting to be served

$$S_t = \langle R_t, D_t \rangle. \tag{3.6}$$

Time introduces special considerations when demands cannot be serviced instantaneously and a time delay restricts observation of action impacts. When time affects the problem model, state space could be modeled to include projected information about resource and demand levels at some point in time in the future. The marshal allocation model employs this approach.

3.4.2. Decision Space and Policy

The literature uses a variety of words and notation to describe decisions, including actions, controls, and policy. This study will refer to $x = (x_d)_{d \in \mathcal{D}}$ as a vector of decisions where $d \in \mathcal{D}$ is an individual decision for a single marshal. The decisions in a MDP are problem-specific but for frame of reference, an individual decision in the context of the marshal allocation problem might be to allocate marshal m to a high-risk flight. Furthermore, π is the overall policy for the model that captures the specific decision vector X_t to execute given state S_t.

$$X_t^{\pi}(S_t) \tag{3.7}$$

The decision vector stores the set of all individual decisions for all eligible marshals at time t of policy determination. This facilitates simultaneous allocations for all eligible marshals at each decision point.

3.4.3. Transition Function

This research focuses on a problem involving stochasticity. Thus, representation of the model environment requires quantifiable randomness as it affects state transitions. Model stochasticity presents as exogenous changes to the state space based on information arriving over the time interval from t to $t + 1$ and is represented by the random variable

W_{t+1} (see Figure 3-3). Randomness can include exogenous changes to the resources $\hat{R}_{t+1}$, new random demands $\hat{D}_{t+1}$, or random changes to other model elements. A sample realization is represented by ω_i where $W_{t+1}(\omega_i)$ is an observation of randomness inherent in the system arriving between time t and $t+1$. The sample can be collected through real-world observation of physical processes, sampling known distributions, or applying computer simulation of a complex process.

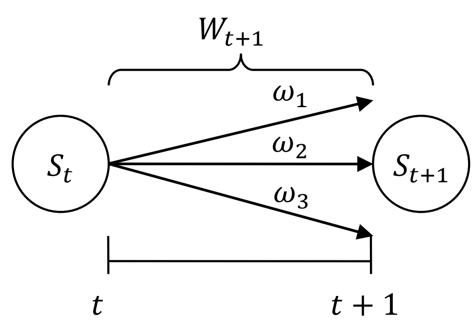


Figure 3-3: State transition model under uncertainty.

A transition model M depicted in (3.8) incorporates the effects of the new information arriving between time steps and describes the evolution of the state variable. The model for evolving from state S_{t+1} is dependent on the current state S_t and the random variable W_{t+1}.

$$S_{t+1} = S^M(S_t, W_{t+1}). \tag{3.8}$$

For ADP learning, the algorithm must also take into account the effects of decisions in conjunction with the random information. If two possible decision vectors exist (x_1, x_2) from state S_t, then the transition in Figure 3-3 is modified as follows:

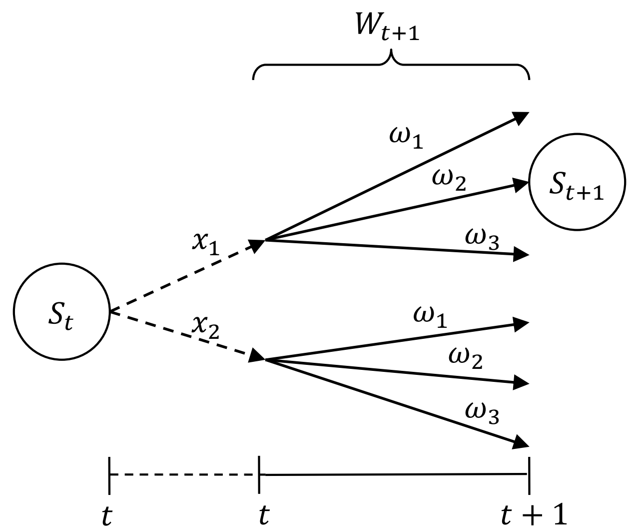


Figure 3-4: Modified state transition model.

The model in Figure 3-4, doubles in size from three branches to six branches with the introduction of two different decisions. Note that there is no passing of time associated with the decision, as the decision is made instantaneously at time t and prior to observing the random variable W_{t+1}. As decision space and outcome space grows so does the problem size. Growth continues until the problem eventually becomes unyielding and requires more innovative solution approaches.

3.5. Post-Decision State

Powell (2011) introduced the use of a post-decision state S_t^x to avoid explicitly solving for the expectations or even requiring transition probabilities. Using a post-decision state (PDS) modifies the transition model in (3.8) into two steps yielding

$$S_t^x = S^{M,x}(S_t, x_t), \tag{3.9}$$

$$S_{t+1} = S^M(S_t^x, \mathrm{W_{t+1}}). \tag{3.10}$$

In equation (3.9) M, x represents the transition model for evolving from the state S_t under decision x_t and brings the system to the post-decision state S_t^x. Note that this is a deterministic transition and occurs without the passing of time. From the PDS the original model M incorporates the random information W_{t+1} and brings the system to the next pre-decision state S_{t+1}. Following this two-step process, Figure 3-4 is modified to that in Figure 3-5.

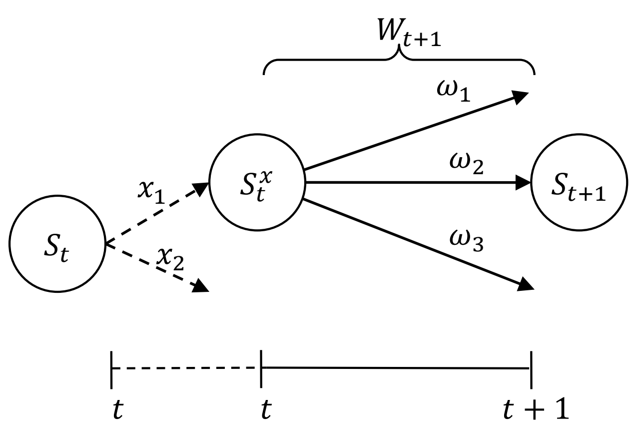


Figure 3-5: State transition model using post-decision state.

Again, take note that there is no passing of time with the decision and that the exogenous information occurs after the decision is executed. While at first glance, the model in Figure 3-5 appears to have the same complexity as the model in Figure 3-4, the states are now modeled around the PDS which makes the problem easier to solve. Furthermore, only the exogenous information related to the selected decision is considered for the future state. Note that the state S_{t+1} is now the next pre-decision state for consideration at the next time step. Powell (2011) exploited a simple relationship between $V_t(S_t)$ and $V_t(S_t^x)$ that is summarized by

$$V_{t-1}(S_{t-1}^x) = \mathbb{E}\{V_t(S_t)|S_{t-1}^x\}, \tag{3.11}$$

$$V_t(S_t) = \max_{x_t \in X_t}\big(C_t(S_t, x_t) + \gamma V_t(S_t^x)\big). \tag{3.12}$$

The result of substituting (3.12) into (3.11) is

$$V_{t-1}(S_{t-1}^x) = \mathbb{E}\left\{\max_{x_t \in \mathcal{X}_t}\big(C_t(S_t, x_t) + \gamma V_t(S_t^x)\big) \ |S_{t-1}^x\right\}. \tag{3.13}$$

Note that in (3.13), the expectation is now positioned outside the maximum operator which will provide considerable computational advantages. Assuming a suitable approximation $\bar{V}_t(S_t^x)$ for the value function around the post-decision state S_t^x then after n iterations of the algorithm the optimization problem is expressed as

$$\hat{v}_t^n = \max_{x_t \in \mathcal{X}_t^n} \left(C_t(S_t^n, x_t^n) + \gamma \bar{V}_t^{n-1}\left(S^{M,x}(S_t^n, x_t^n)\right) \right), \tag{3.14}$$

where $\hat{v}_t^n$ is an estimate of the value of the post decision state $S_t^{x,n}$ and x_t^n is the value of x_t that solves (3.14). Take note that the computation of an expectation is not required as it is captured in the deterministic value approximation function $\bar{V}_t^{n-1}\left(S^{M,x}(S_t^n, x_t^n)\right)$. Value function approximation is discussed in section 3.7. The estimated value $\hat{v}_t^n$ is used to update the value function approximation around the post-decision state according to

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1 - \alpha_{n-1})\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}\hat{v}_t^n, \tag{3.15}$$

where α is the step size used in the learning rate of the algorithm. The learning rate is discussed in section 3.9.2.

A sample algorithm of forward dynamic programming using the post-decision state over an infinite horizon problem (the index t is dropped) is outlined in Figure 3-6.

Step 0. Initialization.

Initialize $\bar{V}^0$, set $n = 1$, and initialize S^1.

Step 1. Choose a sample path ω^n.

Step 2.

2a. Solve

$$\hat{v}^n = \max_{x \in \mathcal{X}^n} \left(C(S^n, x^n) + \gamma \bar{V}^{n-1}\left(S^{M,x}(S^n, x^n)\right) \right)$$

and let x^n be the value of x that solves maximization.

2b. Find the post-decision state

$$S^{x,n} = S^{M,x}(S^n, x^n).$$

2c. Update the value function for the post-decision state

$$\bar{V}^n(S^{x,n}) = (1 - \alpha_{n-1})\bar{V}^{n-1}(S^{x,n}) + \alpha_{n-1}\hat{v}^n.$$

2d. Find the next pre-decision state

$$S^{n+1} = S^M\left(S^{x,n}, W^{n+1}(\omega^n)\right).$$

Step 3. Increment n. If $n \leq N$ go to step 1.

Step 4. Return the value function $\bar{V}^n$.

Figure 3-6: Forward dynamic programming algorithm using the post-decision state

3.6. Curses of Dimensionality

Over the years following the foundation of ADP, various techniques arose to mitigate the problems inherent to large dimensionality. Powell (2011) discusses the challenges posed by large problems in terms of the state space, outcome space, and decision space. He refers to these challenges as the “curses of dimensionality”. First, to address outcome space with unknown transition probabilities (also known as the curse of modeling), a simulation framework generates the Markov state transitions. The simulation approach also supports asynchronous updating of individual states. Synchronous update of all values is infeasible in problems with very large state space. However, a simulation with individual updates results in slower convergence and the need for hundreds of thousands or even millions of simulation iterations. Through simulation-based learning, it is possible under certain conditions to achieve near-optimal (approximate) solutions. To address the inability to store all values for extremely large problems, value function approximation schemes are employed to reduce the storage requirement to a relatively small number of values or parameters.

3.7. Value Function Approximation

For problems of very large problem space, explicit state value storage would be prohibitive. Thus, the literature includes techniques to mitigate the storage challenge. Gosavi (2003) categorized available function approximation methods into three main groups: state space aggregation, function interpolation, and function fitting. Value function approximation addressed the curse of dimensionality by approximating the value of a state such that $V(S) \approx \bar{V}(S)$.

3.7.1. Aggregation

One approach of approximating the state value is to use aggregation. The premise of aggregation is to make a decision, transition to the next state, and then determine the value of the resultant state through an aggregated component of the state variable. For example, if city is an element of the system state, aggregating by region would require storage of far fewer overall system state values. To simplify the problem and computation, an exact solution for an aggregated state space may be disaggregated for an approximate solution to the original, more complex problem.

The disadvantages of the approach also merit consideration. In an aggregated model, potential advantages of the state variable structure are negated, and preservation of the Markov property becomes difficult due to a loss of model fidelity. Even if the Markov property is maintained, solution optimality cannot be guaranteed. Additionally, an algorithm incorporating aggregation tends to lean more heavily on heuristics to mitigate the degraded fidelity of the system state caused by aggregation.

3.7.2. Interpolation

Interpolation also reduces storage requirements by only requiring storage for a smaller subset of state values. Interpolation determines the values for the states that are not stored. Gosavi (2003) described the k-nearest neighbors and kernel-based approaches. When a state is visited that is not stored, the algorithm determines state value by averaging the weighted values of its k-nearest neighbors using weights based on Euclidean distances between state vectors or weights assigned using a kernel-based approach. Interpolation may scale small- to moderate-sized problems. However, interpolation still requires storage, and computational inefficiencies arise from the greater number of value lookups required for each approximation during the evaluation process. For example, an approximation scheme that uses four nearest neighbors to interpolate the value of a state that is not stored requires four separate lookups.

3.7.3. Function Fitting

The final approximation scheme addressed in this research is based off fitting a function based on regression. The general form for value function approximation using regression is

$$\bar{V}(S|\theta) = \sum_{f \in F} \theta_f \phi_f(S). \tag{3.16}$$

where θ_f are the coefficients (parameters) of the basis functions ϕ_f. The literature provides expanded detail on particular forms for basis functions: Fourier approximations, splines, piecewise linear regression, and linear and orthogonal polynomials. However, these techniques require pre-specification of the basis functions and the number of terms in the function. This requirement is avoided through the use of diffusion wavelets.

3.8. Diffusion Wavelets

Diffusion wavelets generalize classical wavelet theory which operates on one- and two-dimensional Euclidean spaces to more general structures such as manifolds. The theory draws on spectral graph theory and identifies orthogonal basis functions from the signals represented in the manifold. As basis functions, diffusion wavelets mitigate the challenges of pre-specification by providing benefits as outlined by Balakrishna (2009):

1. Diffusion wavelets produce the best basis functions in light of a nonlinear and non-stationary value function.
2. The number of features necessary to describe the function is automatically determined using the diffusion operator where information is stored compactly in the coefficients of the basis functions even for multi-dimensional data.
3. The coefficients are updated as part of the decomposition scheme avoiding the need for a separate recursive parameter update scheme.
4. Diffusion wavelets serve as an effective interpolator for values of states not visited during the learning phase of ADP.

By using diffusion wavelets for the marshal allocation problem, the research advances the earlier work of Mahadevan and Maggioni (2006) in applying diffusion wavelets for value function approximations in dynamic programming. A brief review of the technique follows to provide cursory understanding of diffusion wavelet theory extended to this work. A more detailed discussion can be found in the original work by Mahadevan and Maggioni (2006) and Balakrishna (2009). The functional form of this transform in the context of value function approximation is

$$\bar{V}(S|\theta) = \sum_{k=-\infty}^{\infty} c_{(j_0,k)}\phi_{(j_0,k)}(S) + \sum_{j=j_0}^{\infty}\sum_{k=-\infty}^{\infty} d_{(j,k)}w_{(j,k)}(S), \tag{3.17}$$

where $\phi_{(j_0,k)}$ is the scaling function, $w_{(j,k)}$ are the wavelet functions. The coefficients for the basis functions are computed as follows:

$$c_{(j_0,k)} = \frac{1}{m}\sum_{i=1}^{m} \bar{V}_i(S)\,\phi_{(j_0,k)}(S), \tag{3.18}$$

$$d_{(j,k)} = \frac{1}{m}\sum_{i=1}^{m} \bar{V}_i(S)\,w_{(j,k)}(S). \tag{3.19}$$

The orthogonal basis functions allow for perfect reconstruction of the signals. This theory applied to value function approximation in ADP modifies the algorithm in Figure 3-6 to that of Figure 3-7:

Step 0. Initialization

0a. Set size of $\bar{V}$ and $\mathbb{S}$ (sample set of visited states) to $\mathbb{N}$ and initialize $\bar{V}^0$ and $\mathbb{S}^0$.

0b. Set $n = 1$ and initialize S^1.

0c. Set $\mathbb{m} = 1$ and set $\mathbb{s}_1 = S^1$.

0d. *D*efine criteria $\mathbb{D}$ for switching to diffusion wavelet approximator $\tilde{V}$.

Step 1. Choose a sample path ω^n.

Step 2. Check criteria $\mathbb{D}$. If $\mathbb{D}$ is satisfied continue to Step 2a, else go to Step 3.

2a. Obtain/update initial basis functions and the corresponding coefficients for $\tilde{V}^n$ using diffusion wavelets on sample $\mathbb{S}$.

2b. Solve

$$\hat{v}^n = \min_{x \in X^n} \left(C(S^n, x^n) + \gamma \tilde{V}^{n-1}\big(S^{M,x}(S^n, x^n)\big) \right)$$

and let x^n be the value of x that solves the equation.

2c. Update the value of post-decision state $S^{x,n}$:

If $S^{x,n} \in \mathbb{S}$ go to step 4,

else find $\mathbb{s}_{\text{old}} \in \mathbb{S}$ where $\mathbb{s}_{\text{old}}$ is state with oldest recorded visit iteration and replace

$$\mathbb{s}_{\text{old}} = S^{x,n},$$
$$\bar{V}^n(\mathbb{s}_{\text{old}}) = (1 - \alpha_{n-1})\bar{V}^{n-1}(S^{x,n}) + \alpha_{n-1}\hat{v}^n,$$

go to Step 5.

Step 3.

3a. Solve

$$\hat{v}^n = \min_{x \in X^n} \left(C(S^n, x^n) + \gamma \bar{V}^{n-1}\big(S^{M,x}(S^n, x^n)\big) \right)$$

and let x^n be the value of x that solves this equation.

3b. Find the post-decision state

$$S^{x,n} = S^{M,x}(S^n, x^n).$$

3c. If $\mathbb{m} \leq \mathbb{N}$

$$\mathbb{s}_{\mathbb{m}} = S^{x,n},$$
$$\bar{V}^n(\mathbb{s}_{\mathbb{m}}) = (1 - \alpha_{n-1})\bar{V}^{n-1}(S^{x,n}) + \alpha_{n-1}\hat{v}^n,$$
$$\mathbb{m} = \mathbb{m} + 1,$$

else go to step 4.

Step 4. Update $\mathbb{S}$ and $\bar{V}(\mathbb{S})$.

4a. Update

$$\bar{V}^n(S^{x,n}) = (1 - \alpha_{n-1})\bar{V}^{n-1}(S^{x,n}) + \alpha_{n-1}\hat{v}^n,$$

else find $\mathbb{s}_{\text{old}} \in \mathbb{S}$ where $\mathbb{s}_{\text{old}}$ is state with oldest recorded visit iteration and replace

$$\mathbb{s}_{\text{old}} = S^{x,n}$$
$$\bar{V}^n(\mathbb{s}_{\text{old}}) = (1 - \alpha_{n-1})\bar{V}^{n-1}(S^{x,n}) + \alpha_{n-1}\hat{v}^n.$$

Step 5. Find the next pre-decision state

$$S^{n+1} = S^M\big(S^{x,n}, W^{n+1}(\omega^n)\big).$$

Step 6. Increment n. If $n \leq N$ go to step 1.

Step 7. Return the value function approximator $\tilde{V}^n$.

Figure 3-7: ADP algorithm with diffusion wavelet approximation

3.9. Phases of ADP Algorithm

Simulation-based implementation of ADP has three basic phases: exploration, learning, and implementation (or learnt).

3.9.1. Exploration

Exploration occurs at the beginning of the simulation. Initially, the max operator in (3.14) is switched off in place of selecting random decisions. Without the max operator, the system visits states it might not otherwise visit, and the system may identify unlikely neighborhoods exhibiting good state values. Even with the max operator turned off, the exploration phase still takes advantage of the smoothing from the learning rate contained in (3.15). Exploration gradually transitions in favor of the of the maximization operator. With the incorporation of smoothing, an overlap exists between the exploration and learning phases.

3.9.2. Learning Phase and Convergence

The learning phase comprises the entirety of the simulation portion of an ADP strategy. The purpose of the simulation is to achieve optimality of (3.1) by producing an optimal policy for use during the implementation phase. Key considerations during the learning phase primarily concern the relationship between convergence and the learning rate.

The learning parameter α represents a step size and usually takes on values between 0 and 1. The learning parameter is initialized at a high value which places more emphasis, early in the simulation, on the combination of immediate rewards and discounted value of the post-decision state as computed from solving equation (3.14). Towards the end of the simulation as α decays emphasis shifts to the long run value of the state either stored or captured in the value function approximator. Critically, the learning parameter must not decay too quickly nor reach zero prior to meeting convergence criteria, or else learning will cease prematurely. Figure 3-8 graphically depicts a learning rate that exhibits these characteristics.

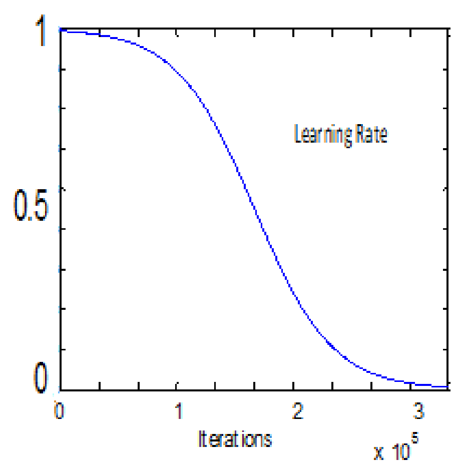


Figure 3-8: Sample learning rate over duration of simulation

By their nature, stochastic approximation algorithms do not converge to a point and therefore necessitate an appropriate convergence scheme. This research chooses to measure convergence based off the mean squared error (MSE) of the gradient. The scheme employed for convergence establishes a moving band based off a ± 1 standard deviation of k prior iterations of the simulation. Figure 3-9 illustrates the MSE decreasing after an initial climb early in the simulation. Zooming in on the plot reveals that MSE fluctuates due to the stochasticity in the system. Observe the MSE plot enter and exit the convergence band several times. This example still requires more iterations to converge within the band.

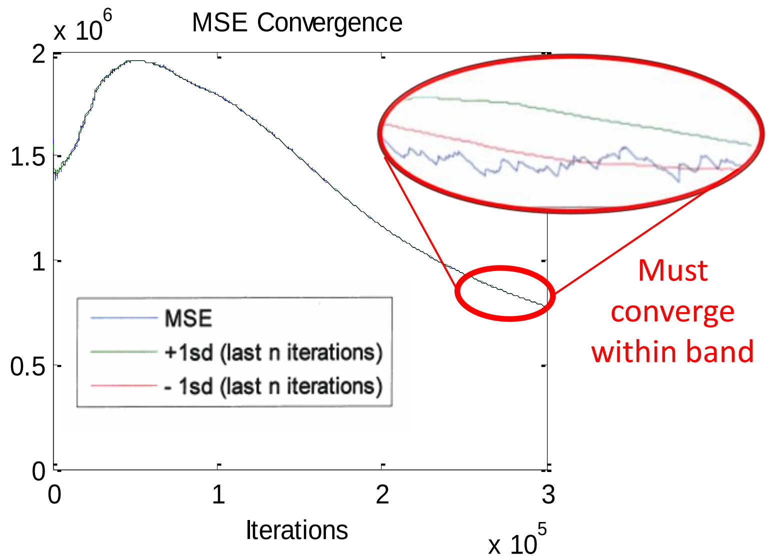


Figure 3-9: Example of convergence to a band

Although the approach is guaranteed to converge, this application requires a significant number of iterations to reach convergence due to the large number of states that must be visited a sufficient number of times accumulate a meaningful value. Further examples of iterations to convergence will be addressed when discussing experimental runs. Two factors address convergence within the approach: 1) step size used for the learning rate closely relates to the convergence of the algorithm; and 2) the approximating scheme for the value function affects convergence but may simultaneously affect ability to converge to optimality.

3.9.3. Implementation (Learnt) Phase

Once the algorithm has successfully converged, the optimal policy is captured in the value function approximator. At this point, all the learning has occurred and policy dictated through the learnt equation can be followed. The learnt policy equation for implementation is

$$x_t = \arg\max_{x_t \in \mathcal{X}_t} \left(C_t(S_t, x_t) + \gamma \bar{V}_t(S_t^x) \right). \tag{3.20}$$

This policy is considered optimal and takes into consideration the downstream effects of immediate decisions. Essentially, the goal of the implementation phase is to move from one good state to another good state by following optimal policy.

CHAPTER FOUR – MODEL FORMULATION

4.1. Overview

This chapter describes the model parameters and data structure along with a description of the dynamics involved in the operating environment. The chapter also discusses how the ADP algorithm and other subroutines interact with the operating environment to learn the decisions that lead to an optimal policy in the long run.

4.2. Inputs and Parameters

To model the air marshals' operational environment, technical parameters address inherent system challenges including size, dynamics, and complexity. This section describes necessary inputs and parameters to capture the system at sufficiently high detail to allocate individual marshals to specific flights. All data used in the model comes from either the public domain or if necessary data is unavailable it is generated during preprocessing phase of the model.

4.2.1. Airports and Flights

The FAA divides the United States into nine regions $r \in \mathcal{R}$ for administrative purposes (see Figure 4-1). Modeling the commercial air transportation of the entire United States has the same difficulty as developing the model for a smaller region. Because the larger scale significantly increases the simulation time required during the learning phase of the algorithm, this research focuses on a case study involving only three of the nine regions. Careful consideration in coding the model environment will ensure easy adaptation to the entire United States. The case study includes only U.S. commercial flights both initiated and terminated at the active airports within the closed system defined by the Eastern, Southern, and Great Lakes regions. The study defines active airports as those departing a minimum of 90 daily flights. The resulting model is a 20-airport system to include 11 major hubs (see Appendix A). Airport data is collected from “Airport Data & Contact Information” (2014) and hubs are classified according to “List of hub airports” (2014).

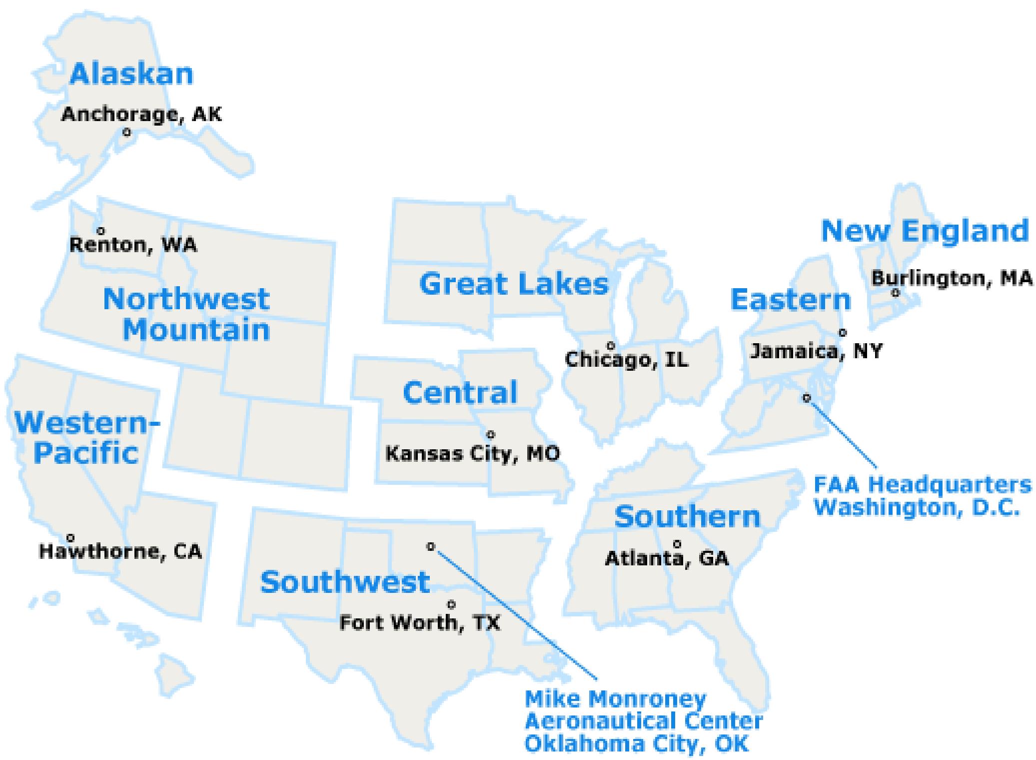


Figure 4-1: FAA regions (“Regions and Aeronautical Center Operations,” 2012).

Hubs are key locations in the system model in that they have more flights departing to more airports throughout the day. This larger number of feasible flights create an opportunity for a broader decision space for marshals. Metropolitan areas with more than one airport in close proximity also create an opportunity for broader decisions: marshals could potentially relocate amongst these airports using means other than the flights in the system. Because modeling metropolitan areas requires a high level of specification, the FAMS may choose to extend the model with this capability after adoption.

The active airports within the defined region support an average of 2,850 flights each day for model simulation. Of the many attributes tracked on all simulation flights, the algorithm uses only eight: a unique flight identifier, departure airport, departure date time group (DTG), arrival airport, arrival DTG, importance, risk level, and coverage status.

$$f = \begin{Bmatrix} \text{flight identifier} \\ \text{departure airport} \\ \text{departure DTG} \\ \text{arrival airport} \\ \text{arrival DTG} \\ \text{importance} \\ \text{risk level} \\ \text{coverage} \end{Bmatrix} \tag{4.1}$$

Flight data and attributes are collected from “FAA Operations & Performance Data” (2014). Historical flight schedules were used covering the period from 1 October to 31 December, 2014. During the learning phase the simulation cycled through the same set of data with new stochastic risks.

4.2.2. Marshals and Field Offices

For planning purposes, the QRF model includes 20 available marshals each day. The set of marshals is depicted as $\mathcal{M}$. Twenty marshals is chosen by assuming the proportional distribution of total number of air marshals across all flights and a fixed percentage of QRF-allocated air marshals. From the 2,000 marshals in force (assumption for the model only, as the actual number is sensitive security information), only 440 proportionally cover the 22% of all U.S. flights restricted to the model regions. After accounting for work schedules, days off, and QRF tour length, an appropriate daily QRF size for the three-region model is 20 marshals. These marshals are domiciled at specified field offices. Of the 20 active airports in the model region, 11 are field offices for domiciling marshals, of which 10 field offices overlap with the 11 hubs.

The simulation tracks approximately two dozen marshal attributes, but a minimal subset of seven attributes (4.2) are required for the algorithm to support decision and policy selection. Remaining attributes support metrics collection and code readability. The seven necessary attributes support the algorithm: a unique marshal identifier, the domicile airport, the QRF duty start date, daily duty start time, allocated flights, projected status, and projected airport. Only the first two are static; the remainder dynamically change based on state transitions. The allocated flights are an array list of flight IDs for current allocations. The projected status and projected airports are vectors discretized by time steps used for forward tracking of the marshal's status and airport, respectively, over the projection period.

$$m = \left\{ \begin{matrix} \text{marshal identifier} \\ \text{domicile airport} \\ \text{QRF duty start date} \\ \text{daily duty start time} \\ \langle\text{allocated flights}\rangle \\ \langle\text{projected status}\rangle \\ \langle\text{projected airport}\rangle \end{matrix} \right\} \tag{4.2}$$

Marshal status space includes six possible statuses: off duty, in-flight, recalled, allocated, available, and inactive. Work conditions constrain marshal availability: a maximum number of consecutive hours worked, a minimum number of consecutive hours off duty, time buffers for boarding and de-boarding flights, and a recall time buffer to travel to an airport after activation. By assumed policy, QRF marshals do not require mandatory off days: after five days on the QRF, marshals return to the regular duty cycle (not modeled). For the QRF, no rule forces the return to domicile at the end of each day. Although marshals may return home each day during the regular duty cycle, policy assumes QRF flexibility to support better assignment by the ADP algorithm.

Other applications of ADP in the literature often aggregate variables to reduce the problem size. For example, to facilitate an optimal ordering policy with ADP, the decisions might be to order different quantity levels of various commodities, within which each item of a particular commodity is identical. In contrast, marshals cannot be aggregated for allocation decisions. ADP as applied to air marshal allocation must define marshals by a number of attributes to uniquely identify each at specific locations in time and space.

4.2.3. Critical Time Periods

Time constrains the model and the decision process. It is found in specified flight arrival and departure times, working rules for marshals, status and location assignments for marshals on flights, and stochastic presentation of risks. This problem considers five significant time intervals, listed in Figure 4-2 from smallest to largest, that are used in the simulation to measure the system state and shape the decision space.

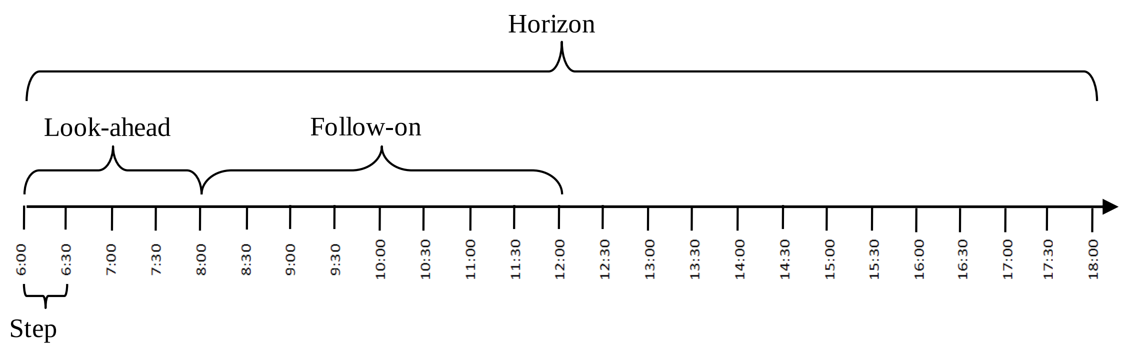


Figure 4-2: Critical time periods.

These time periods are described as follows:

1. **Simulation time step.** Appropriate ranges for values include five minutes to one hour. The size of this interval affects the frequency of policy updates and the storage of marshals' status and location. Larger time steps produce coarser projection of status and location which could exclude some flights from the decision space. Finer discretization by smaller time steps leads to better policy but with diminishing returns. For too small of a time interval, the small measure of entropy between state space transitions could lead to increased iterations for ADP algorithm convergence.

2. **Look-ahead period.** Two hours in length, this period represents the average time to move from the current location to the next as well as an acceptable amount of time to delay in place to cover a known high-risk flight departing later from the same airport.
3. **Follow-on period.** Immediately following the look-ahead period, this four-hour period facilitates decision comparison. The follow-on period provides a set interval to measure the system state after in-flight marshals arrive at their new locations. Critical to the positioning component of the QRF strategy, the follow-on period is when the model measures the potential impact of marshal relocation within the region or rebalance to a new region.
4. **Horizon.** Pre-specified somewhere in the range of 8-14 hours, this time span constrains the value of regional risk as the number of known uncovered high-risk flights. The horizon is a rolling period that shifts with each simulation time step.
5. **One day** (not depicted)**.** The model applies a 24-hour period to determine the number of flights in a particular day and for pre-assigning known high-risk flights to available marshal time (discussed later in this section).

The lengths of these time periods are tunable parameters that may have operational advantages allowing schedulers to align existing procedures with the model. Another advantage could arise from aligning the allocation model to timing aspects of the risk model based on historical observations of data such as ticket purchasing timelines.

4.2.4. Risks

The FAMS may be concerned by two types of risk: static (deterministic) and dynamic (stochastic). Unchanging factors define static risk, such as aircraft size or critical infrastructure along a flight route. Dynamic risk might arise from indicators such as the flight manifest. For example, an individual provides cash for a one-way ticket without luggage. Although not illegal, the suspicious activity may merit additional risk-mitigation by allocating an available air marshal to this flight.

Any of the current approaches can assess both static and dynamic risk. An approach developed by the RAND Corporation assesses risk based on three key components: threat, vulnerability, and consequences (Willis, Morral, Kelly, & Medby, 2005). Another option may apply some metric such as dollars, which would quantify values of assets and human life. A probabilistic assessment component could facilitate calculation of the maximum expected loss (worst case) or the highest probability of loss (most likely case). Risk assessment as applied to the commercial air industry is a sizeable and complex problem with a large number of possible variables and a span covering the entire United States. As such, the scope of this research does not include development of a detailed risk model.

Ultimately, risk is an input to the model. As a simplification of possible risk assessment or measurement methods, the model assumes a binned-risk approach. Furthermore, the model only considers the stochastic risk by assuming that the larger conventional marshal force already provides coverage for deterministic risks. The model uses three risk bins: low, moderate, and high.

$$\mathfrak{R} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \end{Bmatrix} = \begin{Bmatrix} \text{low} \\ \text{moderate} \\ \text{high} \end{Bmatrix} \tag{4.3}$$

The low-risk bin represents a typical flight with no need for an air marshal (unless to reposition the marshal for a later flight). The moderate-risk bin represents an elevated risk level which may warrant a marshal aboard but only after considering tradeoff opportunities. The high-risk bin includes strong readings from risk indicators. The QRF strategy assumes marshals must be allocated to high-risk flights when feasible. The number of flights in each bin responds to the thresholds defined by FAMS' leadership. The model assumes that threshold definitions reduce focus to a limited number of moderate- and high-risk flights with slightly more moderate- than high-risks in the system.

The model only considers perceived risk and excludes risk associated with imminent threat. Any intelligence as to an imminent threat would warrant more drastic measures, such as delaying or canceling departure of the flight. As such, labeling a flight as high-risk does not indicate a direct threat to passengers aboard the flight, but rather, risk bins support the best decisions to allocate a scarce security asset – the federal air marshal.

4.2.5. Policy and Decision Space

A policy is defined as the set of all individual decisions for all eligible marshals at a point in time. Ineligible marshals consist of those still serving their mandatory minimum hours off-duty and those currently in flight; all other marshals are considered eligible and will be included in the optimal policy for that point in time. The set of decisions for marshals (4.4) include: the decision to do nothing (keep marshal in current location), rebalance the system by moving marshal to a hub in another region, relocate the marshal to a hub within current region, activate the marshal for that day, allocate marshal to a moderate-risk flight, allocate marshal to a high-risk flight, abort marshal's currently allocated flight for immediate re-allocation to a high-risk flight, or allocate marshal to a delayed high-risk flight (take lower risk flight leg followed by a high-risk flight leg).

$$d = \begin{Bmatrix} d_0 \\ d_r \\ d_9 \\ d_{10} \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \end{Bmatrix} = \begin{Bmatrix} \text{do nothing} \\ \text{rebalance to region } r \in \mathcal{R} \\ \text{relocate in region} \\ \text{activate} \\ \text{allocate to moderate-risk} \\ \text{allocate to high-risk} \\ \text{abort and reallocate to high-risk} \\ \text{cover delayed high-risk} \end{Bmatrix} \tag{4.4}$$

The feasible decision space for a marshal is defined by that marshal's current status and the flights departing over the look-ahead period from that marshal's current airport. For example, if the set of feasible flights for a marshal does not include moderate-risk departures, then the decision to cover a moderate risk is unavailable. Likewise, if no feasible flights arrive in region r_i, then the decision to rebalance to region r_i is also unavailable. By aggregating decision space, the selected approach enables scalable application to larger problem size. Each aggregated decision includes one or more flights for which the flight selection uses predefined criteria (e.g., taking the flight with highest importance or the flight with the earliest arrival time).

Additional constraints complicate the decision to abort and the decision to cover a delayed high-risk flight. These decisions involve more than one flight and require detailed code specification in the coding to account for sequencing (e.g., a marshal cannot depart on his second leg until after his first leg has arrived). After initial experimentation, addition of these decisions significantly improved performance under all tested strategies. Furthermore, the inclusion of a decision to abort an allocated but not yet departed flight led to an additional modeling constraint: any time a marshal's decision space includes a decision involving allocation to a high-risk flight, that high-risk allocation would automatically be selected for the current policy. The algorithm applies an order of precedence to allocate marshals: first, cover immediate high-risk flight; second, abort current allocation and reallocate to a high-risk flight; and third, cover a delayed high-risk flight.

The algorithm automatically covers high-risk flights as policy and under the assumption that the QRF strategy would never leave a feasible high-risk flight uncovered. The only situation with a tradeoff including a high-risk flight decision involves the possible existence of multiple feasible high-risk flights for a marshal. Testing supports automatic coverage of high-risk flights as long as the decision to abort exists. For example, if a marshal is allocated to a high-risk flight departing in one hour, and a new high-risk flight presents with an earlier departure from the same airport and without other available marshals, then the algorithm would abort the current allocation in favor of the earlier flight. While initially appearing to be a zero sum tradeoff, covering the earlier flight also leaves the possibility that an inbound marshal may arrive in time to cover the later flight.

An example policy π_9 is depicted in Figure 4-3. In this example there are only four eligible marshals for this time decision point (m_7, m_6, m_1, and m_3). The first two marshals (m_7 and m_6) are both assigned decisions involving the allocation to high-risk flights. Marshal m_7 is selected for allocation to a delayed high-risk flight (d_{13}), hence the two flights (f_{15} and f_{376}) associated with the decision, the second of would be the high-risk flight. The second marshal m_6 is assigned the abort decision (d_{12}) in which he aborts current allocation to flight f_{35} and reallocates to the high-risk flight f_{274}. Marshal m_1 is assigned decision d_1 (rebalance to region r_1) which involves allocation to flight f_{456}. Marshal m_3 waits in place (d_0) and has no associated flight. Marhsal m_7

$$\pi_9 = \left\{ \begin{array}{ccccc} \text{marshal} & & \text{decision} & & \text{flight(s)} \\ m_7 & \rightarrow & d_{13} & \rightarrow & \{f_{15}, f_{376}\} \\ m_6 & \rightarrow & d_{12} & \rightarrow & \sim f_{35}, f_{274} \\ m_1 & \rightarrow & d_1 & \rightarrow & f_{456} \\ m_3 & \rightarrow & d_0 & \rightarrow & \text{n/a} \end{array} \right\}$$

Figure 4-3: Example policy for four marshals at time $t = 9$.

There is one final tactic to cover high-risk flights that also measurably and significantly improves performance under all strategies. This tactic is a heuristic that pre-assigns all known high-risk flights over the next 24 hours to available marshal-hours at the beginning of the day. Only performed once a day, assignments under this tactic receive no rewards (the reward scheme for decisions is discussed later in the chapter). This tactic mimics an experienced scheduler as a benchmark to evaluate the performance of the ADP strategy. However, this tactic also resulted in vast improvement of high-risk coverage within the ADP strategy. Because this tactic dictates marshal activation time, pre-definition does not intuitively preclude improvement over dynamic marshal activation time as defined solely by ADP. However, the existence of abort decision proves its power as the abort decision would also apply to any of these pre-assignments. Thus, in addition to representing the skilled scheduler, the algorithm also automatically executes this tactic for all strategies.

4.2.6. State Space

Reliance upon MDP also necessitates definition of possible system states. Careful consideration as to the state representation critically determines the convergence and performance of the ADP algorithm as it learns an optimal policy scheme. The system states include representative features to inform good policy selection and support readable automation in near real-time. Any significant time lag introduced by the information required for state descriptions could inadvertently lead to poor decisions.

Model design must consider the size of the state space. Larger state spaces affect the scalability of the algorithm and require a sophisticated value function approximator when explicit storage of state values becomes prohibitive. Aggregation can be most beneficial to support large state spaces. For example, consider a problem with 100 airports, by which demand is sorted into three risk bins at each airport, and a count of marshals at each airport in three different bins. As an obviously unreasonable number of states, computation requires a prohibitively large number of simulated iterations to visit enough unique states frequently enough to collect a meaningful value. One aggregation approach might bin airports by regions and another might bin on the size of the airports (e.g., hub and non-hub airports). The final version selected for experimentation consists of a vector containing three components. These three components consist of a time block, a means of measuring marshal dispersion throughout system, and a means for measuring uncovered high-risk flights by region in the future.

1. Representative block of time from the day, τ. Flight density varies by time. For example, the density of departures from 6-9 A.M. differs significantly than from 6-9 P.M.
2. Regional airport hubs with feasible coverage, H_r. For inclusion, the hub must have at least one marshal currently located there or inbound within the look-ahead period. This approach captures the distribution of marshals within the region as well as across the regions. By assuming that hubs provide greater connectivity, hubs increase the feasible decision space for marshals.
3. Uncovered regional high-risk flights beyond the follow-on period, U_r. The model does not consider uncovered high-risk in the look-ahead period. If the risk were reachable, a marshal would have already been allocated.

These three components hold information relevant to the decisions in the decision space and will contribute to convergence towards good policy.

$$S = (\tau, H_r, U_r) \tag{4.5}$$

The case study applies six time blocks of equal size. Each region (r_1, r_2, r_3) has a number of regional airport hub values (defined as the total number of airport hubs) and the amount of risk as binned into three categories (no high-risk flights, 1-5 high-risk flights, and 6 or more high-risk flights). The result is a system with 7,776 states for the case study.

4.2.7. Contributions

The contribution scheme penalizes high-risk flights uncovered at each time step and attributes immediate rewards and penalties to individual marshal decisions. The sum of immediate rewards and penalties is captured in the one-step contribution term $C(S^n, x^t)$ found in Bellman's equation. Through recursively updating the state value, this contribution is included in the updated value for the post-decision states. Thus, being in a state with a good decision space (i.e., inclusion of decisions associated with allocating to high-risks) will be rewarded, and the learning algorithm will capture this in the updated value for that state. Conversely, states that result in high-risk flights going uncovered for the immediate time step are penalized, which is also captured in the updated state value. Through simulation, the history of rewards and penalties develops state values which the algorithm will read to inform future decisions.

The reward scheme also evaluates the current distribution of marshals throughout the system. For example, the simulation collects metrics for each critical time period while determining the system state including: the number of marshals currently available at each airport, the number of marshals inbound to each airport during the look-ahead period, and the number of known high- and moderate-risk flights due to depart from each airport. These measurements inform determination of airports and regions with shortages and overages of marshals (i.e., are there enough marshals available to cover known high-risks). Thus, when considering decisions to relocate from one region to another, rewards consider the needs of the originating airport/region as well as those of the destination airport/region. Furthermore, value rewarded for covering moderate- and high-risk flights is scaled by flight importance. Recall that flight importance is a pre-specified deterministic measure stored in flight attribute vector. Flight importance speaks to criticality and vulnerability aspects of risk assessment.

4.2.8. Working Rules

The working rules built into the model consist of constraints on work hours, minimum off-time, buffer times for boarding and de-boarding flights as well as a recall time associated with activating marshals for duty each day.

4.3. Model Dynamics

The model inputs and parameters just described undergo a series of pre-processing steps to insure proper data structuring, validation, and initialization before feeding into the model as depicted in Figure 4-4.

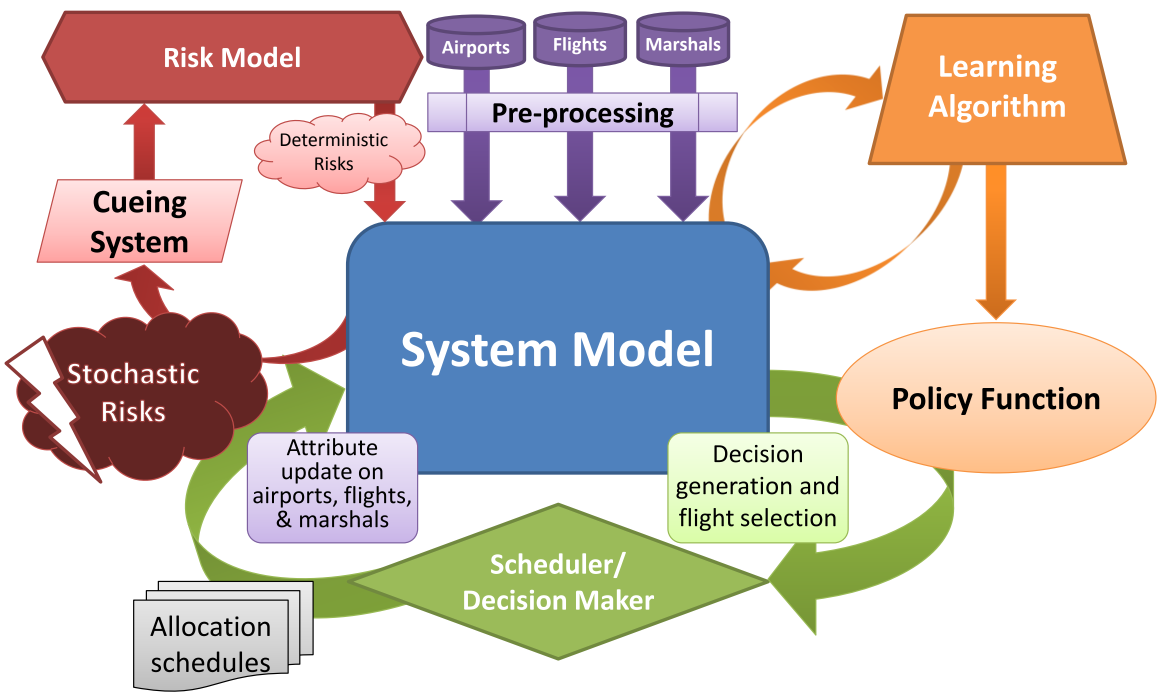


Figure 4-4: FAMS allocation model

The system model consists of the simulation built around the FAMS' operating environment in which flight schedules are realized with the passing of time in discrete time steps. The system makes marshal allocation decisions at each step. In absence of the policy function, a decision maker would have to determine flights for all marshals in the system considering hundreds of flights across dozens of locations. A human would be challenged to consider and factor the immediate impact of decisions, let alone the future impacts or incorporation of stochastic risks. The addition of the stochastic risks makes marshal allocation more challenging because the problem becomes one of balance and positioning to give marshals the greatest chance at covering risks.

In order to arrive at an optimal policy using the ADP algorithm, the simulation must solve smaller sub-problems. One sub-problem involves generating a set of policies for evaluation under Bellman’s equation. One approach to generating such policies involves devising an integer program that would solve to optimality. However, iteratively solving an integer problem of moderate size would be computationally expensive within a simulation-based approach. The simulation runs over hundreds of thousands of iterations as required to facilitate visiting states a sufficient number of times to accumulate a meaningful value for the states. In light of the number of iterations, the algorithm must minimize the computations needed to make decisions on marshals. To minimize the computational effort to evaluate marshal decisions, a heuristic populates the set of policies for evaluation in Bellman’s equation at each time step.

The algorithm employs the heuristic only after populating decision spaces for all eligible marshals. The heuristic makes successive decisions for eligible marshals beginning with the marshal who has the most restrictive decision space (smallest number of decisions). First, if any decision is available to allocate to high-risk flights, it is automatically selected, or else a decision is selected at random from remaining feasible decisions. Next, a check insures that the decisions of preceding marshals do not conflict with the current selection. For example, two marshals at the same airport at the same time most likely share overlapping decision space. If the first marshal is allocated to the only flight supporting a particular decision, then that decision would no longer be available to the second marshal. In such a situation, the selected decision for the latter marshal is reversed and replaced with another random decision from the remaining decision space and re-checked. If the selected decision is still feasible, then a flight is intelligently selected using another heuristic from the remaining flights supporting that decision.

The flight selection heuristic uses criteria to rank flights and selects the highest-ranked flight. Rules define the criteria and may include such relative comparisons: higher importance, earliest arrival times, ending at a high-demand airport, and so on. The next step after flight selection is calculation of the immediate contribution associated with the individual marshal decision. The immediate contribution is a combination of rewards earned from the reward scheme which rewards certain decisions and scales rewards by flight importance. Additionally, arrival at in-demand locations receive rewards, and arrival at locations with an excess of marshals is penalized.

After going through all eligible marshals, the algorithm identifies and penalizes any high-risk flights that will go uncovered over the next time step. The algorithm determines the chosen state based on the combined effect of all marshal decisions, the immediate rewards from the marshals, the immediate penalization of missed flights, and the discounted value of the following state. This resultant value of the policy accounts for both the immediate rewards from marshal decisions as well as the value of the post decision state (the value of the post decision state holds information about the stochastic element which will not be observed until after implementation of the policy).

The algorithm evaluates all policies in the generated set of policies for that iteration. Application of Bellman's equation to all polices defines the policy resulting in the maximum value – this policy becomes the selected policy for that iteration. Using the set of individual marshal decisions and the flight assignments previously determined during policy generation phase, the simulation updates attributes accordingly. This heuristic performs exceptionally fast for a relatively small number of generated policies. As the number of policies to evaluate increases, the computational effort increases for determining the PDS value using the value function approximation whether through a lookup table or other means such as diffusion wavelets.

4.4. Attribute tracking

After every iteration in the simulation marshal and flight attributes must be updated based upon the decisions made, flights selected, and the realization of exogenous information. Updating the flight attributes is straight forward as it entails only the changing of risk level and coverage status. However, tracking the status and location of marshals proves challenging especially in light of decisions that involve allocation to multiple flights or aborting existing flights in favor of another. To handle these particular dynamics, a marshal's availability is projected out 24 hours from the current time and tracked in discretized time steps. For example, if a marshal is allocated to a flight then the relevant time steps in that marshals projection vectors (status and location) are changed accordingly. See Appendix B: Sample Marshal Attribute Tracking for sample marshal attribute tracking.

4.5. Special Considerations

Measurement of effectiveness must maximize the coverage of high-risk (or weighted risk) rather than maximizing utilization of marshals on high-risk flights. While some correlation exists between the two definitions of effectiveness, the two distinct metrics each have different purposes. Utilization speaks towards the efficiency of marshals (e.g., whether his work day is filled with mission hours as opposed to waiting at an airport all day). Effectiveness relates to whether or not marshals are allocated to the flights possessing the types of risk they are designed to cover.

Since risk is binned as a model input, the modeler may set the threshold on the high-risk bin such that many high-risk flights exist. An abundance of high-risk flights provide the opportunity for marshals to fill allocation schedules with high-risk flights and support an appearance of excellent utilization and effectiveness. Conversely, the modeler can set the threshold on high-risk flights so low relative to the number of marshals that any scheduler could easily cover all high-risk flights and provide the perception of effective coverage. Both cases are trivial and not the intent of achieving optimized policy. Recall that every effort should be made to cover a high-risk flight while moderate-risk flights pose trade-off opportunities. With the marshal allocation rules and guidance from organizational leadership in mind, the modeler must carefully tune flight risk bins to insure maximal coverage of the right types of flights.

4.6. Sample Marshal Routes

Consider that marshals are constrained by a set of working rules such as a maximum number of hours to work in a day or having a minimum number of hours off prior to starting a new shift. Given this information, a marshal's available time on a given day is limited, and marshals are likely to cover only 2-4 legs in a given day. This number of legs is further constrained by flight times and cross-regional flights which diminish a marshal's remaining available hours. The following examples demonstrate possible utilization rates on high-risk flights given a non-trivial scenario where the number of high-risk flights and available marshals are appropriately matched.

4.6.1. Two-Leg Work Day

Figure 4-5 depicts a marshal allocation schedule of 2 legs for the day. In such a scenario, a marshal at airport A either starts his daily shift with a known high-risk flight departing from his location. Alternatively, he waits in place until a future flight's risk changes to high, at which point the marshal is allocated to the high-risk flight taking him to airport B. B may or may not be a hub airport. From airport B, the marshal can either wait to see if a moderate- or high-risk flight (depicted by amber and red dashed lines, respectively) presents, or relocate to a hub or marshal's domicile. Factors contributing to only 2 legs might be cross-regional flight, long flight duration, or a long wait time at either airport.

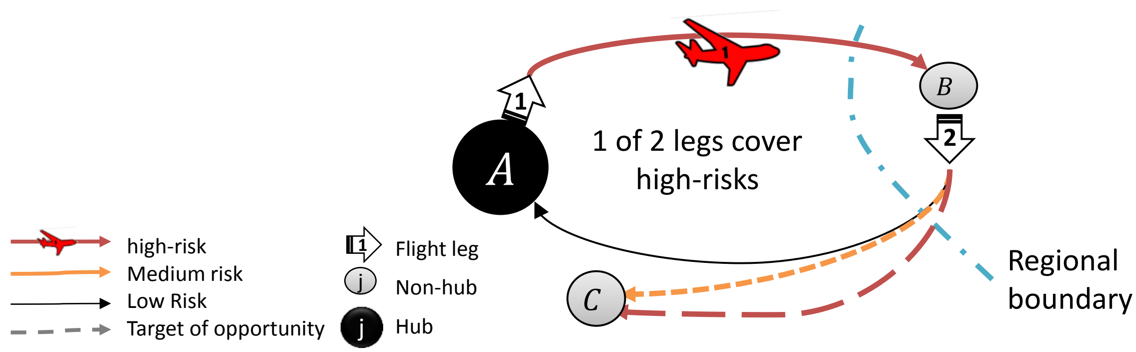


Figure 4-5: Two-leg work day

Note that in this scenario, only 50% of the marshal's legs cover a high-risk flight. Only by chance of another high-risk flight presenting at airport B in the appropriate time window will both legs cover high-risks.

4.6.2. Three-Leg Work Day

Figure 4-6 depicts a 3-leg work day. In this scenario, a marshal starts his work day at airport A (probably a hub if he had the opportunity to reposition at the end of the preceding day) and is allocated to a delayed high-risk. To cover the delayed high-risk, the marshal takes a low-risk flight from airport A to airport B and then boards the high-risk flight destined for airport D. Again, airport D may or may not be a hub. At airport D, the marshal finds himself in the same position that the 2-leg marshal found himself in after the first leg. Either by chance a higher risk flight presents offering an opportunity to cover a second high-risk flight, or the marshal repositions or returns to his domicile for optimal placement during next work shift.

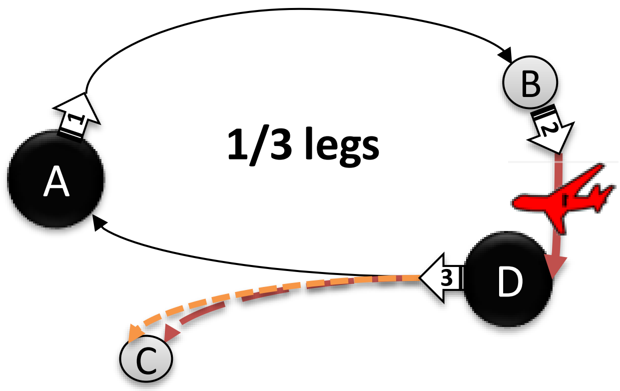


Figure 4-6: Three-leg work day

As in the 2-leg example, only 1 of the legs covers a high-risk flight unless by chance, a flight presents departing from airport D in the appropriate time window.

4.6.3. Four-Leg Work Day

The last scenario covers a marshal able to fit 4 legs in his shift. In such a scenario, the marshal likely stays in the same region and covers short flight legs. In this scenario, the marshal boards a high-risk flight departing from his starting airport much like in the 2-leg scenario. However, from airport B a known high-risk exists such that the marshal may feasibly cover a delayed high-risk departing from airport D. Finally, the marshal has enough time to reposition at the end of the day if airport C is a non-hub or is not a high-demand airport.

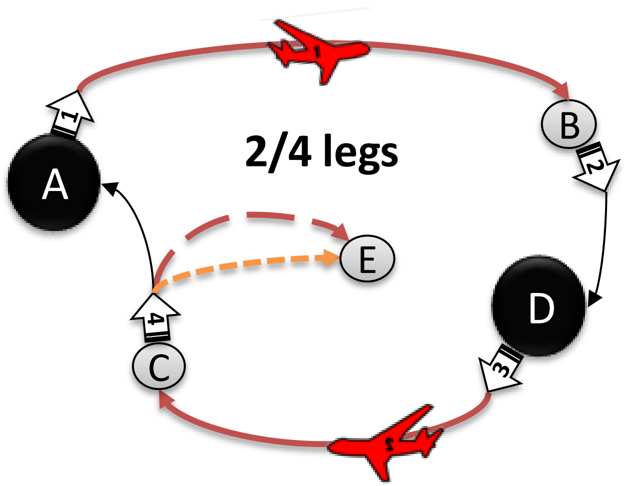


Figure 4-7: Four-leg work day

As a result, 50% of legs cover high-risk flights. These three examples depict cases where marshals cover high-risk. A scenario also exists in which a marshal is not well-positioned or makes trade-off decisions that result in missed opportunities to cover the stochastic high-risk flights. Measured by flight legs in each of the three cases, each marshal's daily utilization on high-risk flights is at or below 50%, barring exposure to high-risk by chance. Proper tuning of the risk thresholds should result in utilization rates just below 50%. The utilization rate will serve as a good starting point for the ADP algorithm to seek optimal policy and exceed these rates by covering more high-payoff stochastic risks through careful consideration of trade-offs and positioning marshals for the greatest opportunity to cover maximal risk.

CHAPTER FIVE – EXPERIMENTATION & ANALYSIS

5.1. Experiment Design

Experimental design addresses the non-trivial scenario in which the definition for high-risk flights tunes the number of potential flight allocations to the number of available QRF marshals. Recall that confidentiality limits access to existing procedures and metrics. Thus, the research requires design of a reasonable and appropriate means of measuring performance suited to the experiments.

Each experiment run in the research applies the same time period and same number of simulations. The experiments are run for 10 day periods over 25 simulations. All parameters are held constant except for those distinguishing the strategy being tested. For sensitivity analysis, only those parameters explicitly being examined will vary.

5.1.1. Controls

The experiment uses two controls: the myopic strategy and simulation of an experienced scheduler following a heuristic. These two strategies are described below.

5.1.1.1. Myopic Strategy

To establish baseline performance, the research uses a basic myopic strategy for comparison. The myopic strategy follows an identical decision and reward scheme as the ADP algorithm except that it gives no consideration to the state value of the system after a decision cycle. Mathematically, the difference is equivalent to setting the discount parameter in Bellman's equation equal to zero. Thus, the modified one step optimization problem becomes

$$V(S^{x,n}) = \max_{x^n \in \mathcal{X}^n} C^n(S^n, x^n) \tag{5.1}$$

Implementing this modified equation at every iteration of the simulation has the effect of taking a short-sighted greedy approach that only considers the immediate rewards for decisions made at the current time and ignores the potential impact of future decisions. As this strategy requires no learning phase, the myopic approach results in the immediate implementation of the policy that satisfies the maximization problem

$$x^n = \arg\max_{x^n \in \mathcal{X}^n} C^n(S^n, x^n) \tag{5.2}$$

Note that once the ADP algorithm learns the optimal policy, simulation no longer serves a purpose. As a result, indexing on equation (5.2) would revert to time t rather than iteration n. However, in experimentation the simulation model serves to test the controls as well as the other strategies by building a distribution around the performance metrics.

5.1.1.2. Experienced Scheduler

The second control is to mimic the procedures an experienced scheduler might follow. To implement the control, a subroutine pre-assigns known high-risk flights over a specified time period to marshals' available time over the same period. This subroutine extends from the decisions for individual marshals: allocate to high-risk flights and allocate to delayed high-risk flights. After pre-assigning all known high-risks flights to marshals, the scheduler adopts the myopic strategy for the duration of the time period. At the conclusion of the time period, pre-assignment executes for the next time period. A logical time period for upfront pre-assignment is the 24-hour hour period starting at midnight. This time period covers the entire on-duty and off-duty cycle for most marshals.

The experiment assumes perfect information for the experienced scheduler: all would-be stochastic risks are populated up front with scheduler visibility on those risks 24 hours out. By treating these risks statically, the experiment captures how a scheduler would perform with perfect information. The performance of the scheduler under perfect information serves as a pseudo-upper bound or goal to be reached by any other strategy under testing.

Take note that the pre-assignment subroutine proved to be a powerful heuristic that improved performance of all tested strategies. In conjunction with the existence of an abort decision, pre-assignment provides immediate benefit. With an abort decision, a marshal may abort any pre-assigned flight if a better opportunity presents stochastically. To incorporate the improvement, the ADP model also includes this heuristic. The benefit of pre-scheduling is an important observation of the research process demonstrating the overlap of the modeling and experimentation phases and the iterative means under which the research process occurred. This is an excellent example of the emphasis that an ADP approach places on the modeling aspect of the problem more so than other OR approaches.

5.1.2. Risk

The experiments use seeded sequences to model stochastic risk such that each strategy encountered identical risks throughout. The base scenario establishes the threshold for the moderate-risk bin at 5% and the threshold for the high-risk bin at 1%. The lead time for visibility of risk is ranged with a maximum of 12 hours and a minimum set to the length of time used for the boarding buffer (e.g., 30 minutes). The maximum lead time represents a situation such as a potential terrorist purchasing a ticket with a half day notice. The minimum lead time represents a potential terrorist purchasing a ticket at the last available moment. The lead time for all risks is uniformly distributed across this time range. This is not to suggest that the risk propagation would occur in this manner, but uniform risk distribution forms a basis to demonstrate algorithm performance. Recall that the risk model is an input to the system. Should the FAMS adopt the model, they would likely employ their expertise to incorporate a more accurate and detailed representation of risk.

5.1.3. Metrics

The simulation tracks a number of metrics for performance against the environmental model, including high-risk coverage, weighted coverage, and marshal utilization on high-risk flights. High correlation exists among these metrics. Functional instances may arise where one metric more meaningfully captures system effectiveness, especially to planners and schedulers. For the purposes of this research, the high-risk coverage metric provides an option which is simple to calculate and explain while addressing a key assumption: the FAMS must make every opportunity to cover high-risk flights.

5.2. Experimental Runs

The main experiment consisted of setting the baselines on a consistent scenario: 20 airports support a daily average of 2,850 flights with approximately 1% posing high-risk and 5% posing moderate-risk. Risks randomly emerge with visibility uniformly distributed between 12 hours before departure and immediately prior to boarding. System variables result in an average of 29 high-risk flights each day to be covered by 20 marshals. Each marshal starts his QRF tour of duty at a pre-specified field office. Due to the relaxed working rules assumed for the QRF, the marshals are exempt from the requirement to return to domicile at the end of each day. Should a marshal complete QRF duty away from his domicile, he will return to his domicile on an administrative flight. A secondary objective of the main experiment is to test the viability of using diffusion wavelets for approximating the value function used in the ADP algorithm and the learnt equation.

The results of this experiment are illustrated in Figure 5-1: Mean coverage of high-risk flights (measured against myopic strategy).. The three gray bars in the chart depict the performance of the ADP learnt policy against three approaches (from left to right): a straight value lookup approach with no approximation, an interpolation scheme on a reduced-size lookup table, and diffusion wavelets as the approximation scheme. As a positive finding, both approximations methods performed within statistical bounds of the base ADP model with no approximation. Other trial runs with various parameter settings produced consistent results.

With respect to performance the ADP policy covers slightly more than 60% of the stochastic high-risk flights on average that appear over the simulation. In comparison to the myopic strategy (46.3% coverage) is an absolute improvement of almost fifteen percentage points or equivalently 4-5 high-risk flights on this scaled version of the problem. Consider the full-scale problem and this improvement could amount to the coverage of dozens more high-risk flights.

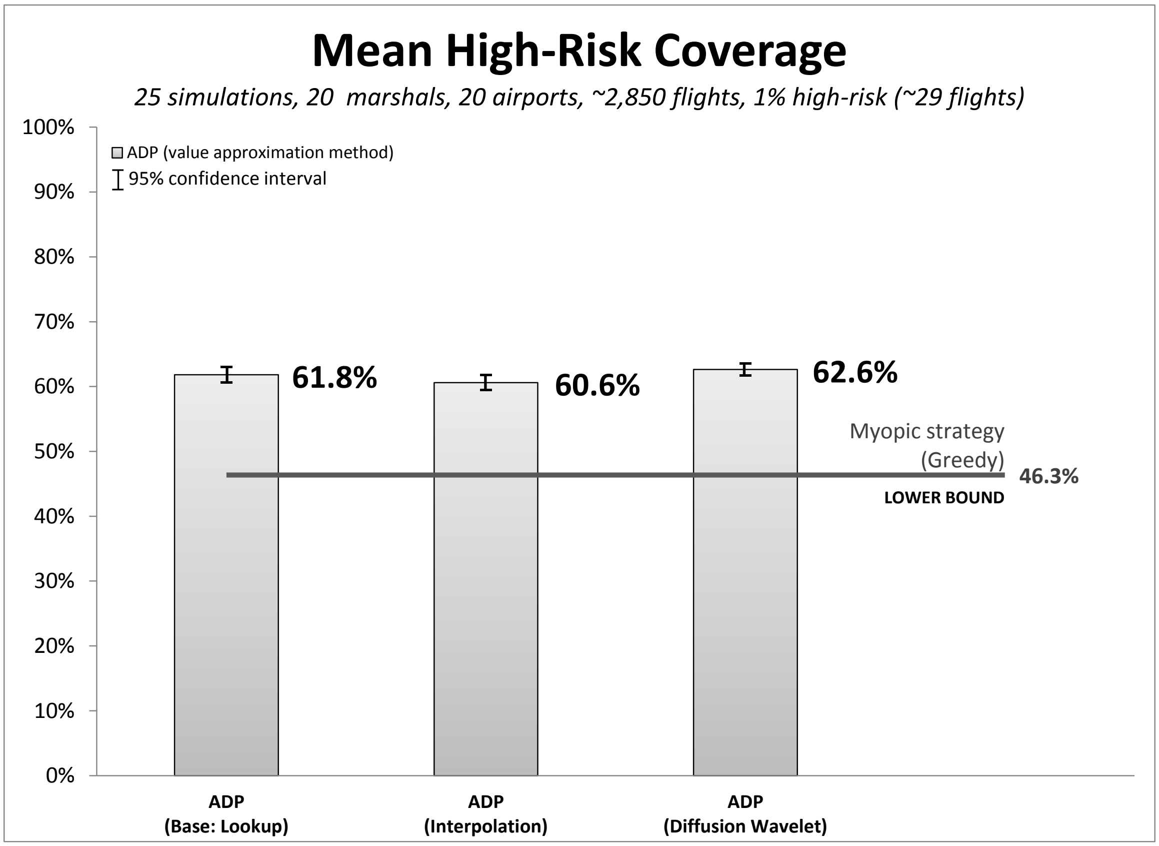


Figure 5-1: Mean coverage of high-risk flights (measured against myopic strategy).

The next step in this experiment is to compare the performance of the ADP policy to the pseudo-upper bound captured by modeling the experienced scheduler operating under perfect information. Figure 5-2 reveals that the experienced scheduler covers, on average, 71.4% of the high-risk flights given a QRF of 20 marshals and a high-risk threshold set to 1% of the daily flights. Despite the deterministic nature of risks under perfect information scenario, the experienced scheduler does not produce optimal results: he is merely following a heuristic that mimics how a human might go about scheduling. An integer program (see Appendix E: Integer Programming Formulation for an example integer program) could easily solve to optimality the deterministic version of this problem over a finite time horizon, but this experiment only employs comparative measures appropriate for sequential decisions under uncertainty.

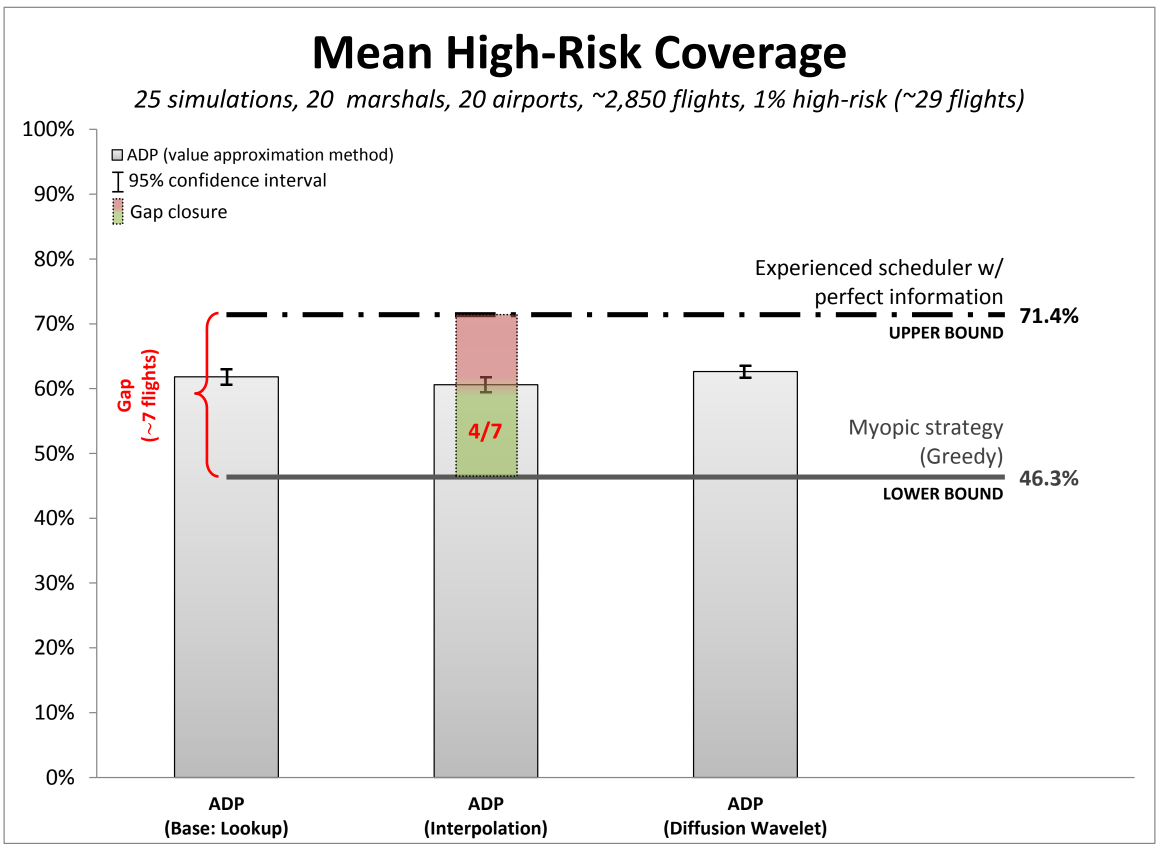


Figure 5-2: Mean coverage of high-risk flights (measured against experienced scheduler under perfect information).

This chart highlights the gap between the myopic strategy and the experienced scheduler under perfect information. The flights in this gap comprise the flights that may benefit most from an effective allocation strategy. These high-risk flights emerge with very little lead time and at locations with no marshals present. The key observation is that the ADP strategy covers more than half of these flights (4 out of 7). This finding demonstrates the power of the ADP policy in light of stochastic risks.

For the approximately 29 daily high-risk flights in the experiment, on average, the myopic strategy covered between 13 and 14, ADP covered between 17 and 18, and the experienced scheduler under perfect information covered between 20 and 21 high-risk flights. Although the experiment tests a limited number of high-risk flights, the scaled model only includes 3 of 9 regions, covering approximately 22% of domestic flights (meeting criteria for inclusion) in the United States. Additionally, this strategy applies only to the QRF, assumed to be only 10% of the FAMS' force at large. Finally, and perhaps most importantly, only a very small percentage of real-world flights will meet criteria for a high level of perceived risk arising from individual passengers.

5.3. Sensitivity Analysis

Sensitivity analysis explored two parameters: the number of marshals and the number of high-risk flights. By choosing these two parameters, the research tests algorithm performance as the system approaches the boundaries of the non-trivial scenario. The scenario becomes trivial when marshals begin to greatly outnumber risks and vice-versa. Understanding the limits of the marshal and flight parameters will aid planners in defining the high-risk flight threshold responsive to the number of QRF marshals.

5.3.1. Amount of Risk

The first sensitivity test explores the realized improvement of ADP as the amount of risk in the system varies. Fixing the number of simulated QRF marshals at 20, the percent of flights with high-risk varies: 0.5%, 1%, 2%, 3%, and 4%. Preliminary tests include all three variants of ADP strategy used in the main experiment. Relative comparison among the ADP variants reveal consistent performance without statistical significance, as in the main experiment. Thus, the research only presents the results of ADP using interpolation as the value function approximations scheme. The research employed interpolation as the strategy variant because it required the shortest amount of simulation run time. Figure 5-3 depicts the results from the analysis.

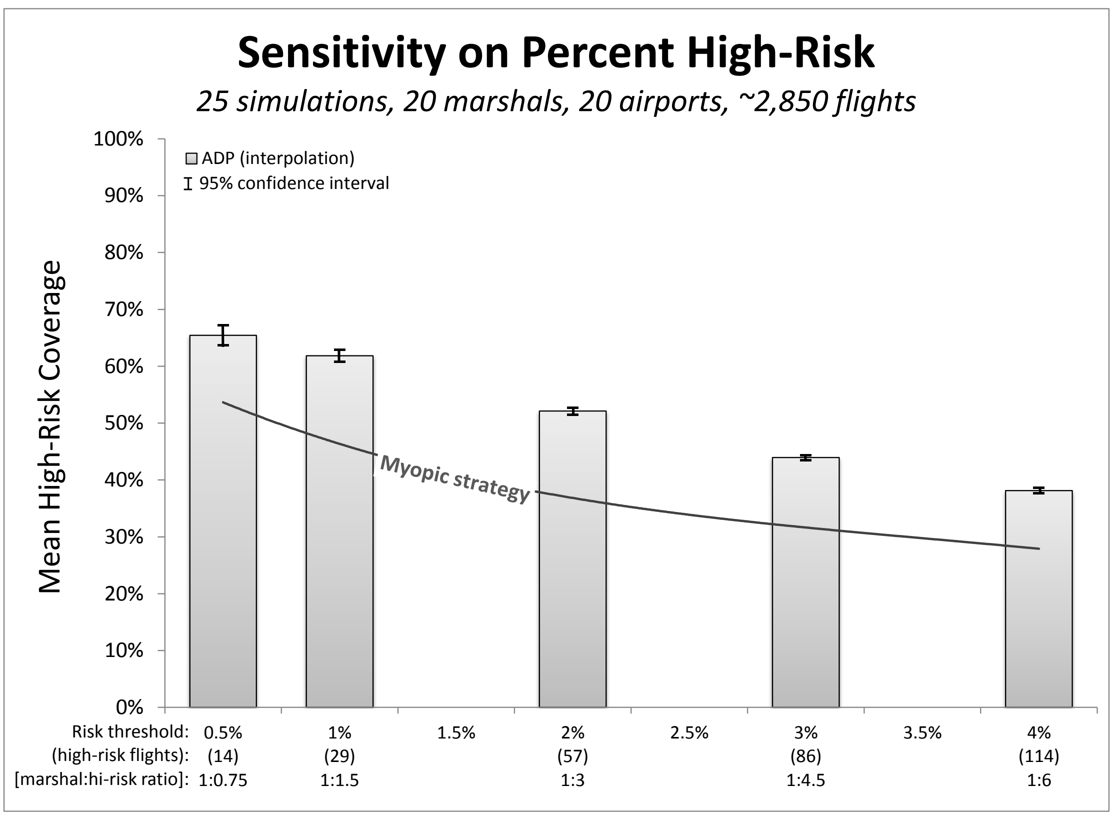


Figure 5-3: Sensitivity on percent of high-risk (measured against myopic strategy).

As a key observation from the analysis, the ADP algorithm consistently outperforms the myopic strategy. Another observation is that high-risk coverage decreases as the high-risk threshold increases. By fixing the number of marshals at 20 as high-risk flights increase, the marshal availability limits the total daily capacity for potential high-risk coverage.

As in the main experiment, the experienced scheduler under perfect information sets the pseudo-upper bound which caps the basis for the critical gap. The additional comparison provides valuable information in Figure 5-4: Sensitivity on percent of high-risk (measured against experienced scheduler under perfect information).. It is important to note that although the gap size appears to be decreasing (moving from left to right), the number of high-risk flights comprising the gap simultaneously increases due to the increased number of high-risk flights in the trial runs. This analysis reveals that at all parameter settings tested, the ADP approach covered an average of 50% or more of the gap flights.

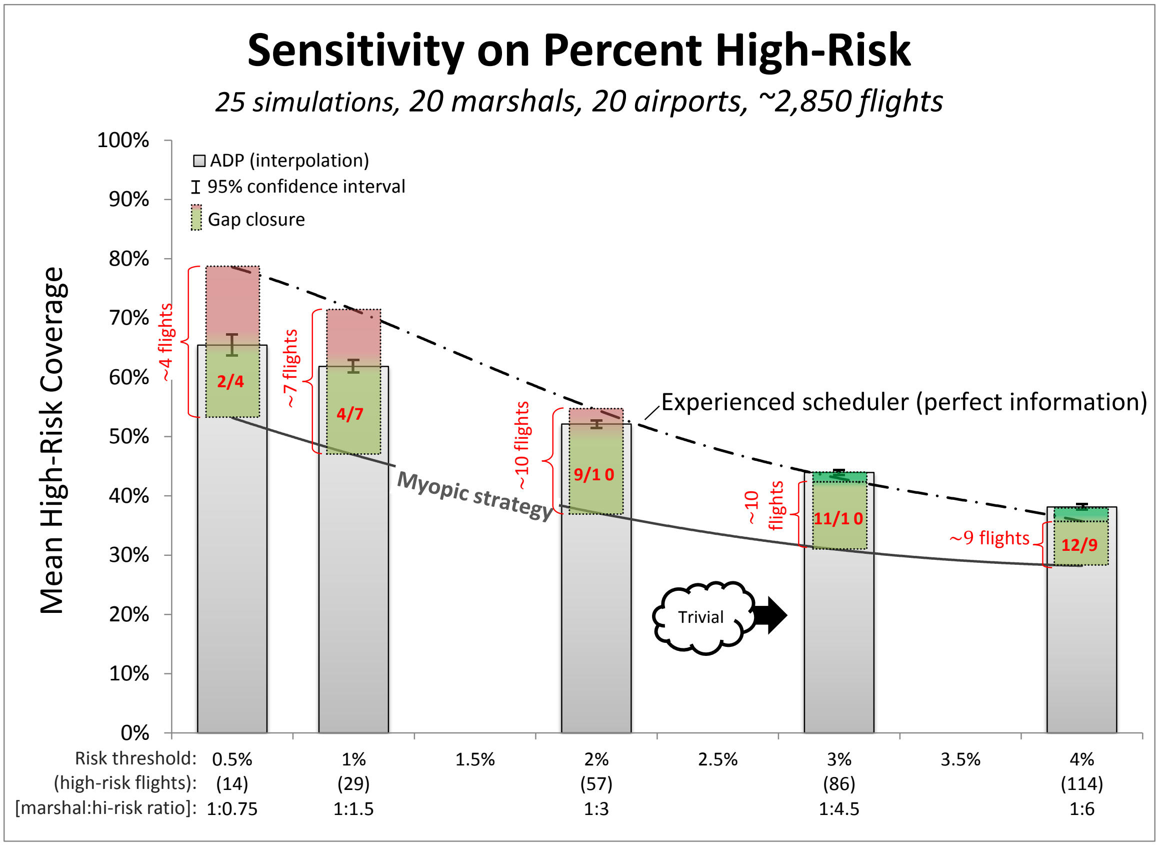


Figure 5-4: Sensitivity on percent of high-risk (measured against experienced scheduler under perfect information).

The analysis also reveals an increasing trend of gap coverage. However, as the number of high-risk flights exceeds three for every marshal, the system becomes trivial: many more high-risk flights than marshals produce high marshal utilization for high-risk flights. As a consequence from setting the high-risk threshold too high, high-risk flights become indistinguishable to the algorithm. In reality, some of the high-risk flights might be more desirable to cover than others. There are two ways to target the more desirable high-risk flights. Firstly, a more sophisticated risk model paired with a reengineered state vector, but this would result in an even larger state space. Alternatively, careful tuning of the high-risk threshold may create the desired effect. Optimal tuning may involve analyzing historical observations of risk cueing systems, adjusting the number of QRF marshals available, and/or setting an acceptable level of exposure to uncovered high-risk flights. This latter method is an intuitive finding resultant from the experimentation and is recommended as the preferable approach.

Improved ADP performance for rising system risk (relative to the pseudo-upper bound) suggests two things: 1) the ADP algorithm performs exceptionally well at balancing and pre-positioning QRF marshals throughout the day to cover more stochastic risk when an abundance of risk persists, and 2) a high-risk threshold greater than 2% encroaches on the boundary for which this application becomes trivial. In the trivial instance of weakly defined high-risk, so many high-risk flights are available that marshals fill their daily allocation schedules with mostly high-risk flights. Recall that these are not imminent risks but perceived risk based on positive indicator signals.

5.3.2. Number of Marshals

The second sensitivity test explores the number of marshals in the system. The simulation fixes the percent of high-risk flights at 1% (average of 29 daily high-risk flights) while testing varied numbers of QRF marshals: 10, 20, 40, 60, and 80. As in the previous sensitivity analysis, discussion only includes the results of the ADP algorithm using interpolation as the value approximating method. Figure 5-5 shows the results from the analysis.

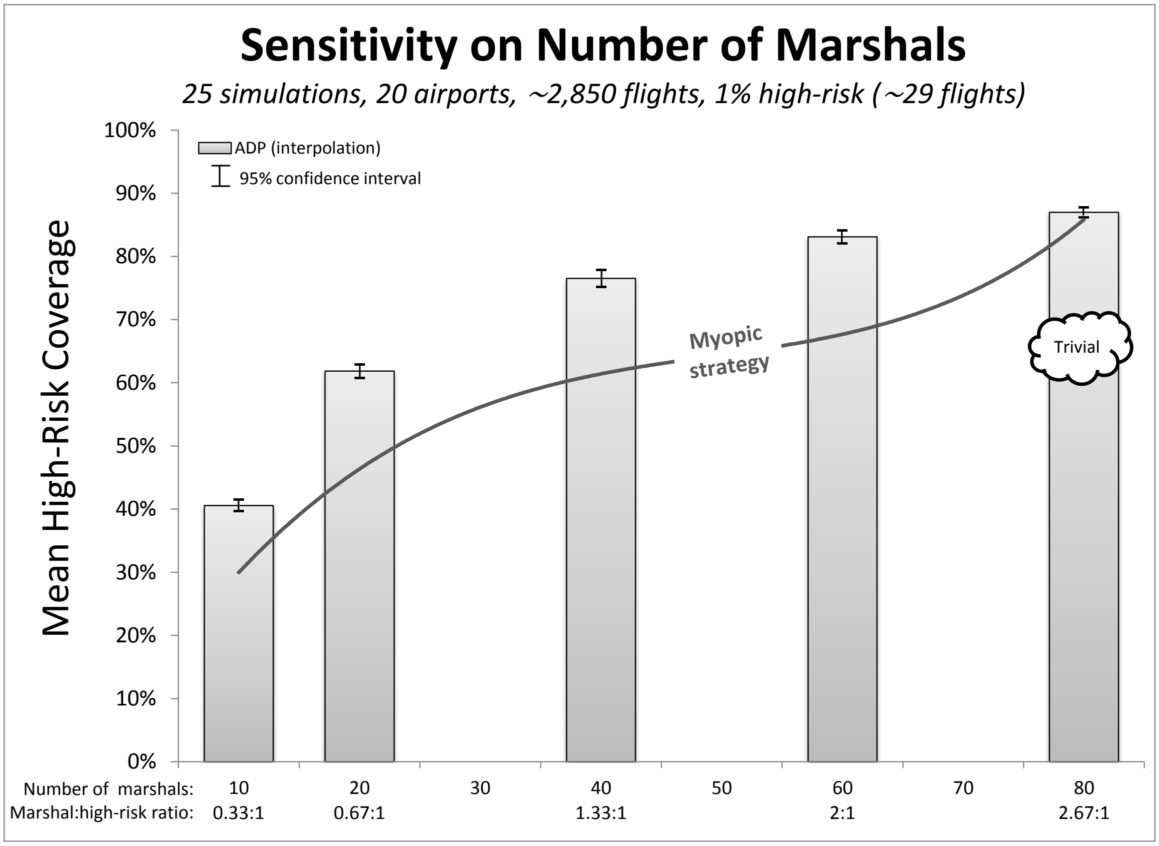


Figure 5-5: Sensitivity on number of marshals (measured against myopic strategy).

The analysis reveals that increasing the number of marshals on the QRF leads to increased coverage of high-risk flights but with diminishing returns. Additionally, the ADP strategy clearly outperforms the myopic strategy for all instances of the number of marshals. However, when the marshal to high-risk flight ratio approaches 2.67:1 (80 marshal scenario), the negligible improvement possibly indicates the boundary for the trivial scenario. With many more marshals than high-risk flights, marshals easily cover most high-risks flights regardless of allocation strategy used.

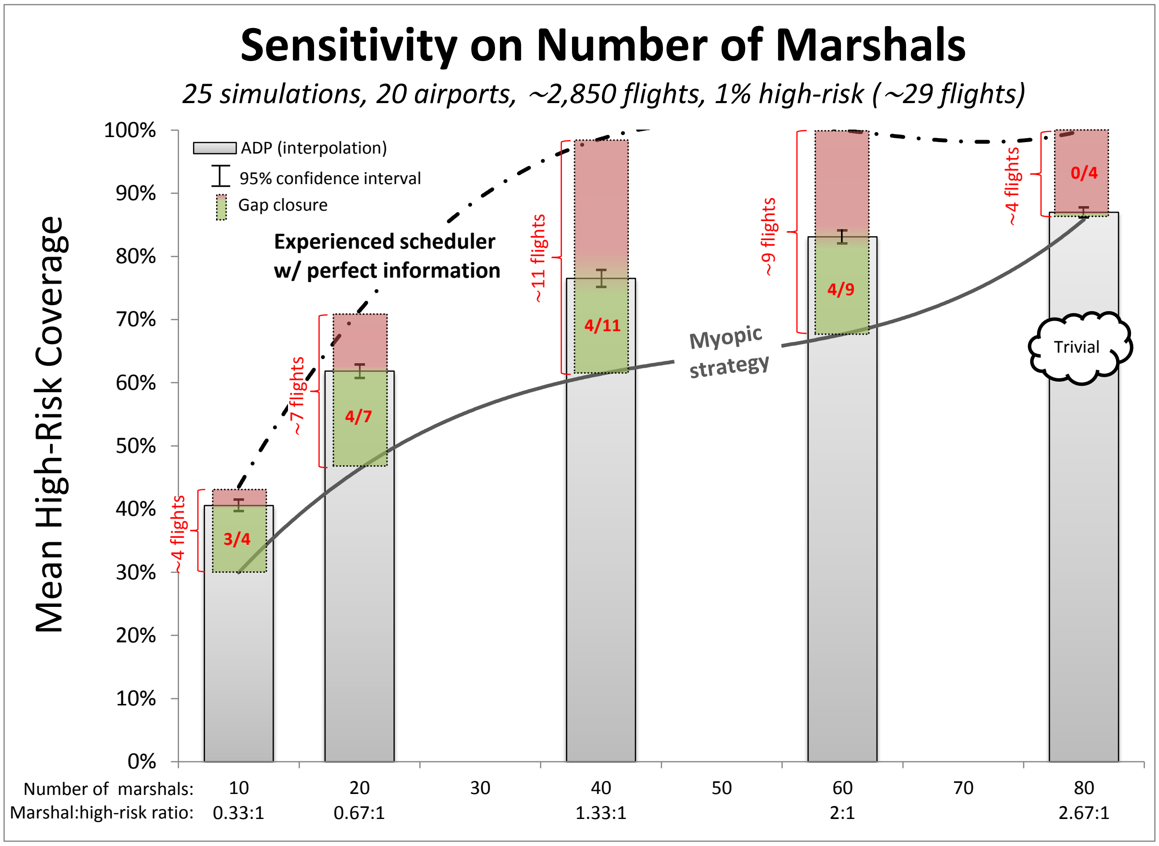


Figure 5-6: Sensitivity on number of marshals (measured against experienced scheduler under perfect information).

The most significant observation from testing numbers of marshals is that the ADP algorithm closes the gap between the myopic strategy and the pseudo-upper bound as defined by the experienced scheduler under perfect information, but with diminishing returns. The largest gap closure (hence, the greatest benefit from ADP) occurs for 10 marshals and results on average in coverage for 3 of the 4 high-risk flights in the gap. At 20 marshals, ADP also performs well by covering more than 50% of the gap (4 of 7 flights). However, observe that the gap generally increases until reaching the trivial scenario, and ADP benefit diminishes as more marshals become available in the system. As the marshals begin to outnumber the high-risk flights in the scenario, the experienced scheduler easily covers all high-risk flights given perfect knowledge of the risk. As the number of marshals increases, the myopic strategy improves coverage more slowly than the experienced scheduler because the stochastic element remains in the myopic approach. The widening gap explains the diminishing returns on ADP gap closure as the system moves from 10 to 80 marshals.

5.4. Observations on Problem Size

This case study includes a 3-region subset of the FAMS area: the Eastern, Southern, and Great Lakes regions of the United States. Relative to the 9 regions that comprise the entire U.S., the selected regions support approximately 22% of total flights in the FAMS system. Problem size impacts both modeling considerations and simulation performance.

5.4.1. Modeling

Table 5-1 compares the scale of main variables as modeled and in actuality. Note that although the case study uses one third of the FAMS regions in the United States, the model includes fewer than 22% of the daily flights. The significant difference in relative percentages arises from modeling the area as a closed system: the model only includes flights that both depart and arrive within the 3 selected regions. Model constraints exclude many of the cross-regional flights such as coast-to-coast flights that a full, 9-region system would include.

Table 5-1. Problem size comparison.

Problem Size	Small	Large	Scaled Size
Regions	3	9	33.3%
Airports	20	72	26.7%
Mean Daily Flights	~2,850	~13,000	21.9%
QRF Marshals	20	100	20.0%

In comparison to the 3-region case study, the 9-region system exhibits potentially exponential growth of state space. Note that in Table 5-1, the size of the state space for the modeled problem relative to the potential problem changes by many magnitudes of order. In the small problem, all state values may be explicitly stored, regardless of whether or not the problem approach includes aggregation. However, the vast size of the state space of the larger problem obviates the impossibility of explicit storage supported by a prohibitive demand for computer memory and processing time associated with lookup routines performed on such large tables.

Table 5-2. State space comparison.

State Size	Small	Large
Aggregating on region	$> 10^4$	$> 10^{13}$
No aggregation	$> 10^6$	$> 10^{22}$

The implications of scale changes for model adoption clearly highlight the benefits of aggregating the model as well as the greater need for and importance of incorporating value function approximation as problems size scales.

5.4.2. Computation

The true benefits of value function approximation arise during the learning phase of ADP, which requires many iterations to accumulate meaningful state values. As such, simulation run times can span hours, days, and even weeks depending on the size of the model and the method of value approximation employed. Table 5-3 illustrates the run time differences for the modeled and actual problem space in a scenario applying state-space aggregation at the regional level.

Table 5-3. Computational comparisons of learning phase.

Simulation Run Times (Aggregated Model)	**Small**	**Large**
No value function approximation	< 8 hrs	--†
VFA: interpolation	< 6 hrs	days (estimated)
VFA: diffusion wavelets	< 24 hrs	weeks (estimated)

†Storage capacity prohibitive.

Note that for the small case study, the optimal policy can be learnt in under a day, and possibly in only a few hours, regardless of the approximation scheme chosen. The full-scope model, with the exception of the lookup table approach, could take weeks or even months to learn, depending upon the system state-space definition. Exceptionally long run times arise without value-function approximation due to the ever-increasing size of the lookup table. Each time the application visits a new unique state, the lookup table incorporates the value and adds to the computational time required to perform future lookups. The diffusion wavelet scheme could also take weeks, with computational time driven by complex matrix operations performed on moderately sized matrices multiple times for each simulation iteration. Note that the code used to perform the diffusion wavelet approximation in the experimentation bears room for improvement, and efficiencies could reduce the run time from weeks to days. Future work on the application may address efficiency reduction for diffusion wavelets.

Long run times for the learning phase should not ward off the end-user of this technology: the learning phase need only occur once (or as warranted by large perturbations to the system or model). Once learning occurs, ADP transitions to the learnt policy equation for implementation. The implementation phase experiences far less drastic change in scale between models. Regardless of system size, any approach will only take a matter of seconds to determine the optimal policy given the current state of the system. Table 5-4 presents the run times for the aggregated model for all three variants of ADP strategy.

Table 5-4. Computational comparisons of implementation phase.

Implementation Run Times (Aggregated Model)	**Small**	**Large**
No value function approximation	< 1 sec.	n/a
VFA: lookup table	< 1 sec.	seconds
VFA: diffusion wavelets	< 1 sec.	seconds

Note that QRF implementation to address stochastic high-risk assumes a risk-evaluation system exists. The risk-evaluation system would need the ability to relay threat assessments of all flights over the specified critical time periods in near-real time such as that discussed in the literature review. Additionally, the status and location of all marshals must be dynamically updated and tracked in near-real time to serve as an input into the learnt equation.

CHAPTER SIX – CONCLUSIONS

6.1. Findings

In following the theme of contributions, which parallel both methodology and application, this chapter provides conclusions from the research on each of these facets.

6.1.1. Methodology

The primary theoretical objective was to apply ADP to influence decision making under uncertainty in a domain area exhibiting scarce resources and low-frequency stochastic events. Problems exhibiting these conditions require careful consideration to the positioning and balance of resources to effectively be able to react to the infrequent stochastic events. This work shows significant improvements of the ADP approach compared to myopic strategies that seek to maximize short term benefits and ignoring the value that may arise in the future by intelligently repositioning assets according to an optimal policy learnt through simulation approach.

The secondary theoretical objective of the research was to address scalability by demonstrating diffusion wavelet as a suitable value function approximator. Very few researchers have explored diffusion wavelets as use as value function approximator in ADP. Thus the successful implementation in this application serves to promote its use and advance the scalability of ADP on problems sizes previously considered too big to solve.

This work shows diffusion wavelets performed on par with other approximation techniques such as aggregation and table lookups.

6.1.2. FAMS Application

The application objective was to demonstrate the potential benefits from allocating federal air marshals in near real-time according to optimal policy derived from ADP. This research demonstrates that ADP is a viable approach for learning optimal policy to facilitate near real-time allocation of federal air marshals to counter a stochastic risk. Not only does the approach facilitate synchronous decisions on multiple marshals, but it does so in a matter of seconds. The implementation phase is data driven by a small number of inputs. In addition to the policy equation, these inputs include updated flight schedules, statuses for all available QRF marshals that day, and latest risk assessments on all flights.

The first input for the policy equation, is likely already accessible to the FAMS. Near real-time updates to flight schedules via their close coordination with TSA and individual airlines should be of little difficulty to attain. Marshal statuses for daily availability – to include all the tracked attributes – could be updated through a merger of existing marshal tracking procedures and the subroutine used in the simulation for updating marshal attributes. Thus, this aspect of model is attainable, too. However, the last element, the risk assessment, poses some challenges. Ideally, existing systems such as Secure Flight already monitors and collects information requisite to make a valid risk assessment based off stochastic indicators in near real-time. If this were not the case, the system would have to be augmented with potentially new technology to provide the necessary assessment capability of the factors that would differentiate between the risk bins. Obviously this last requirement is the limiting factor in realizing this capability. This research could serve as justification for pursuing such technology that can perform the risk assessments without fear over privacy concerns that plagued CAPPS II.

Regardless of the existence or development of risk assessment technology for assessing passenger manifests in near real-time, the algorithm has potential to complement existing procedures for scheduling FAMS against more deterministic factors. During the research experimentation phase it was shown that the model can be applied to deterministic risks – as was the case with the experienced scheduler operating under perfect information. The underlying model used for the algorithm can produce feasible schedules for a day, weeks, or months, even for large numbers of marshals in a relatively quickly manner. The building of the schedule would occur dynamically using existing deterministic risk models fit to model structure and the heuristics already built into the model. This approach avoids the long solution times associated with solving large integer programming problems providing schedulers with quicker solutions. This approach is also vastly different than traditional means of crew scheduling problems in which subsets of feasible flight pairings are generated and considered for inclusion in the solution of crew scheduling. Such a technique only considers a relatively small portion of feasible pairings possible out of the essentially infinite combinations. The proposed model considers all feasible flights accessible to a marshal from his immediate location over the look-ahead period as well as follow-on flights over the follow-on period. This has the potential to find stronger pairings, those with multiple moderate- to high-risk flights paired together.

Additionally, the model can also serve as a scenario analysis tool for planners. The model has the potential to track details of the system at a very high level of detail that can help planners make policy decisions on numbers of marshals to domicile at particular airports, to help de-conflict activation times for marshals at the same airport or operating in the same region, to test performance of different scheduling heuristics, to collect marshal utilization statistics for analysis, and to visualize the distribution of marshals and risks across time and space. Some of these capabilities are built into the model and ready to go while others require only slight modifications to adjust to actual procedures and data structure.

Throughout the research process, discussions arose with committee members and other analysts as to ways to game such an approach along with proposals to counter gaming attempts. For security reasons these discussions are not included as part of this published work but will be shared with the FAMS as a part of future collaborative efforts to improve security operations and homeland defense.

6.2. Broader Impact

This research serves to advance scalability limitations of employing ADP on large problems by demonstrating diffusion wavelet as a suitable value function approximator. Diffusion wavelet theory can significantly reduce the storage requirements down to a relatively small number of parameters yet still facilitate modeling at a high level of detail.

The employment of this approach for near real-time scheduling of QRFs is not limited to the FAMS but could apply to any defense or security agency that employs a QRF strategy consisting of multiple entities disbursed over time and space where the allocation to a demand of one entity may offset the balance of the system requiring temporary rebalancing or repositioning. Furthermore, the concept is not restricted to QRFs but can be adapted to other large systems requiring constant rebalancing and repositioning in order to maximize the likelihood of servicing stochastic demands.

6.3. Future Work

On the algorithmic front there are areas where the model and algorithm could perform better and faster. Some examples include exploring more sophisticated learning rate schemes that are more adaptive to the convergence gradients of the simulation. Improving model robustness by including additional stochastic elements such as flight delays and cancelations as well as personnel absences and tardiness. An improved risk assessment scheme that factors in the remaining portion of unsold seats on a flight. Including this in the model would affect the probability of specific flights changing to a higher risk level (e.g., if a flight is sold out and already assessed at the lowest stochastic risk level than there is a zero probability of that flight changing to a higher stochastic risk level). This component of modeling would contribute to reducing some of the uncertainty in the model.

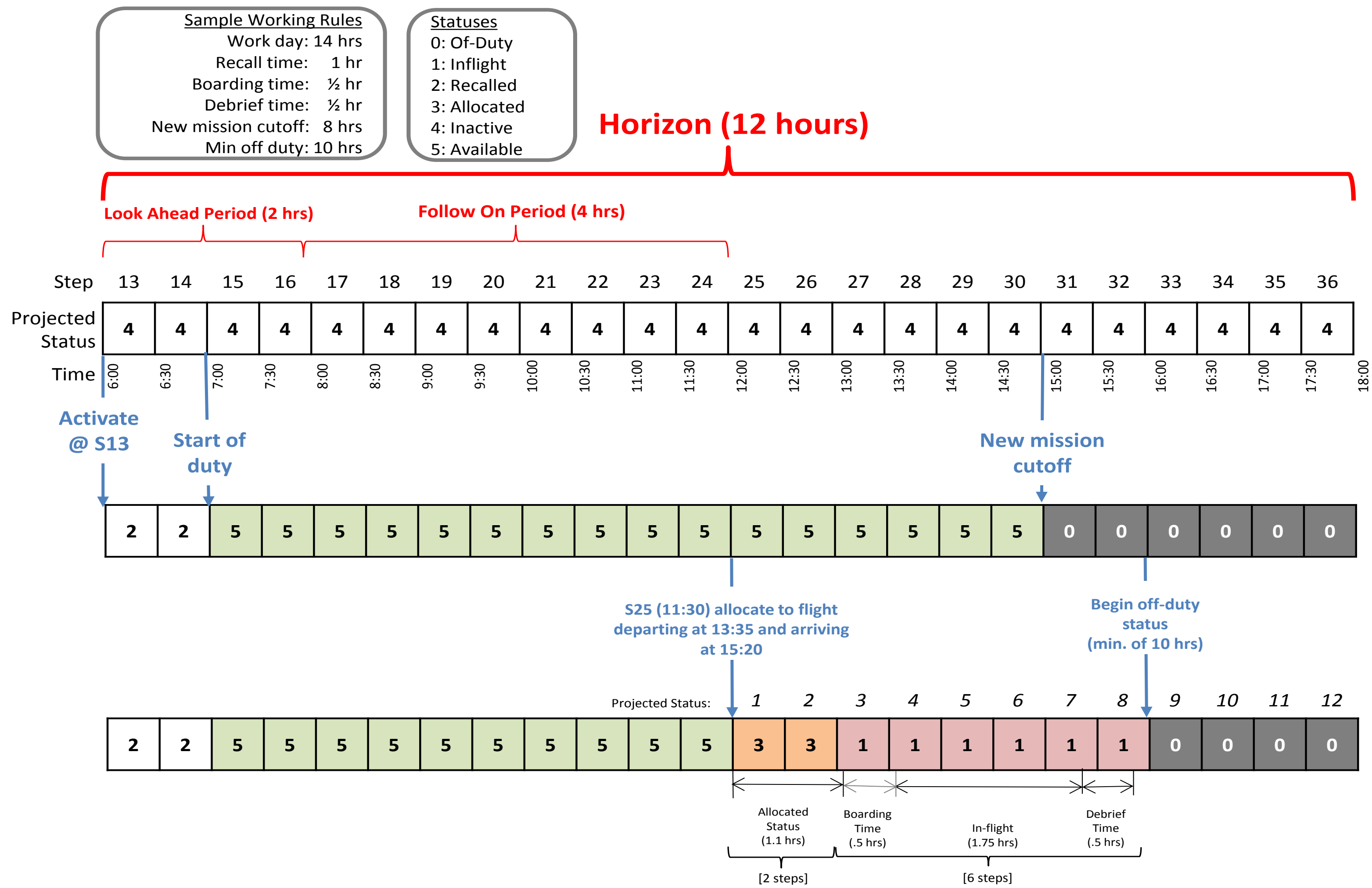
Of course, the most important future work to come is through collaboration with the FAMS, adapting the model to their existing data structure, and finally testing the model performance against existing procedures. Findings from these tests might warrant pilot testing and/or merging the approach with existing procedures to form more robust scheduling and allocation procedures that effectively address all sources of risk.

APPENDIX A: AIRPORTS IN THREE-REGION MODEL

#	Region Index	Region Name	Airport Index	Airport Code	Airport Name	Airport City	Hub
1	*1*	*Eastern*	*25*	***BWI***	*Baltimore/Washington International*	*Baltimore, MD*	*Yes*
2	*1*	*Eastern*	*39*	*DCA*	*Ronald Reagan Washington National*	*Washington, DC*	*No*
3	*1*	*Eastern*	*50*	***EWR***	*Newark Liberty International*	*Newark, NJ*	*Yes*
4	*1*	*Eastern*	*71*	*IAD*	*Washington Dulles International*	*Washington, DC*	*No*
5	*1*	*Eastern*	*78*	*JFK*	*John F. Kennedy International*	*New York, NY*	*No*
6	*1*	*Eastern*	*89*	*LGA*	*LaGuardia*	*New York, NY*	*No*
7	*1*	*Eastern*	*152*	***PHL***	*Philadelphia International*	*Philadelphia, PA*	*Yes*
8	*1*	*Eastern*	*153*	***PIT***	*Pittsburgh International*	*Pittsburgh, PA*	*Yes*
9	*2*	*Great Lakes*	*270*	***DTW***	*Detroit Metro Wayne County*	*Detroit, MI*	*Yes*
10	*2*	*Great Lakes*	*362*	***MDW***	*Chicago Midway International*	*Chicago, IL*	*Yes*
11	*2*	*Great Lakes*	*396*	***MSP***	*Minneapolis-St Paul International*	*Minneapolis, MN*	*Yes*
12	*2*	*Great Lakes*	*419*	***ORD***	*Chicago O'Hare International*	*Chicago, IL*	*Yes*
13	*3*	*Southern*	*525*	*ATL*	*Hartsfield-Jackson Atlanta International*	*Atlanta, GA*	*No*
14	*3*	*Southern*	*536*	***BNA***	*Nashville International*	*Nashville, TN*	*Yes*
15	*3*	*Southern*	*554*	***CLT***	*Charlotte Douglas International*	*Charlotte, NC*	*Yes*
16	*3*	*Southern*	*601*	*FLL*	*Fort Lauderdale-Hollywood International*	*Fort Lauderdale, FL*	*No*
17	*3*	*Southern*	*695*	*MCO*	*Orlando International*	*Orlando, FL*	*No*
18	*3*	*Southern*	*702*	*MIA*	*Miami International*	*Miami, FL*	*No*
19	*3*	*Southern*	*776*	***RDU***	*Raleigh-Durham International*	*Raleigh/Durham, NC*	*Yes*
20	*3*	*Southern*	*837*	*TPA*	*Tampa International*	*Tampa, FL*	*No*

*__BOLDED__ airport code = hub airport

APPENDIX B: SAMPLE MARSHAL ATTRIBUTE TRACKING



APPENDIX C: SAMPLE ALLOCATION SCHEDULE

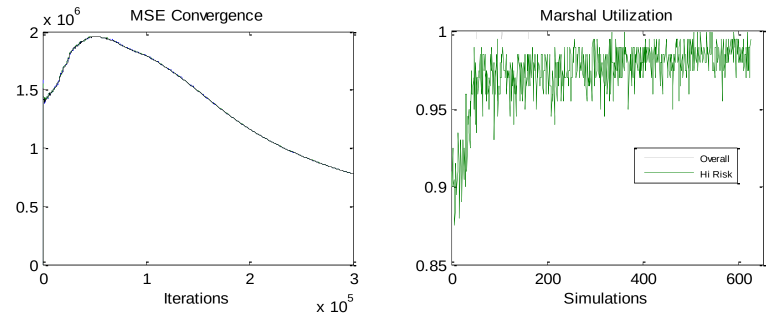
The below schedule is a representative example of the output from the model, grouped on marshal ID and sorted by hour of decision. The white section to the right contains attributes relating to the marshal and the gray section to the left depicts assigned flight attributes.

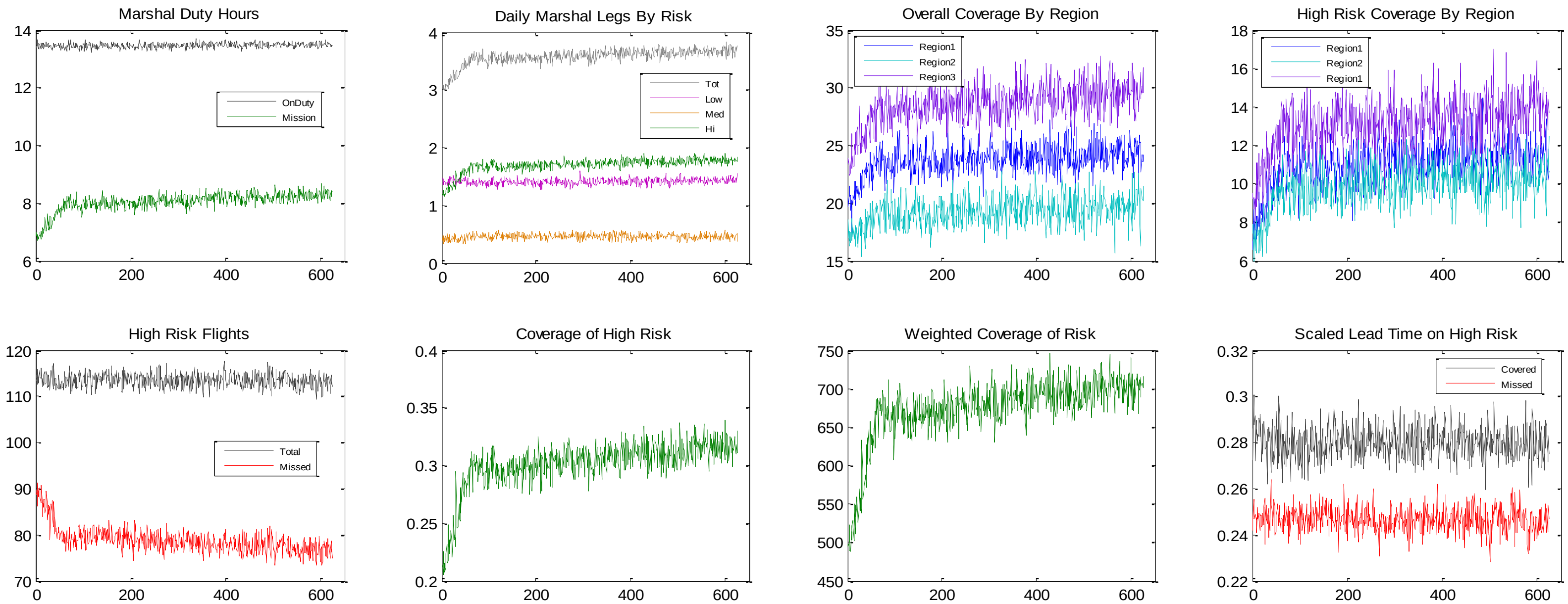
Hour of Decision	Marshal ID	Current Region	Current Airport	Decision	Hrs On Duty (Pre-Decision)	Daily Mission Hours (Post-Decision)	Cum Legs (Mod-erate)	Cum Legs (High)	Legs Total	Flight ID	Depart Hour	Arrive Hour	Arrival Region	Arrival Airport	Risk	Flight Importance
5.50	1	2	270	Activate	0.00											
6.50	1	2	270	Relocate to Region 3	0.25	1.53	0	0	1	26344	10.27	11.80	3	776	Low	1.3
12.50	1	3	776	Cover Moderate	6.25	2.57	1	0	2	26989	13.97	15.00	2	419	Moderate	1.3
15.50	1	2	419	Cover High	9.25	5.65	1	1	3	27700	17.85	20.93	1	50	High	1.4
7.00	2	1	153	Activate	0.00											
8.00	2	1	153	Delayed High (leg 1)	0.25	0.50	1	0	1	26289	10.08	10.58	2	419	Moderate	1.3
8.00	2	2	419	Delayed High (leg 2)	0.25	~~2.10~~ 0.50	1	~~1~~	~~2~~	~~27517~~	~~16.85~~	~~18.45~~	~~2~~	~~270~~	~~High~~	~~1.3~~
10.50	2	2	419	Abort & Cover high	2.75	3.18	1	1	2	27416	16.27	18.95	1	39	High	1.5
19.50	2	1	39	Cover Moderate	11.75	4.02	2	1	3	28224	21.08	21.92	3	776	Moderate	2
7.00	3	1	50	Activate	0.00											
8.00	3	1	50	Delayed High (leg 1)	0.25	1.47	0	0	1	26153	9.22	10.68	2	419	Low	1.8
8.00	3	2	419	Delayed High (leg 2)	0.25	2.70	0	1	2	27280	15.50	16.73	1	152	High	1.3
17.00	3	1	152	Cover Moderate	9.25	4.05	1	1	3	27883	18.72	20.07	3	776	Moderate	1.3
7.50	5	3	536	Activate	0.00											
8.50	5	3	536	Cover Moderate	0.25	2.98	1	0	1	26712	12.27	15.25	1	50	Moderate	1.4
15.50	5	1	50	Relocate in Region	7.25	4.20	1	0	2	27398	16.22	17.43	1	153	Low	1.8

APPENDIX D: SAMPLE SIMULATION METRICS

Key Simulation Parameters

- Strategy: ADP
- 300,000 iterations
- 625 ten-day cycles
- 2,850 flights (4% risk)
- 20 airports
- 20 marshals





APPENDIX E: INTEGER PROGRAMMING FORMULATION

The formulation below is a simplified representation of the deterministic version of the allocation problem for a finite time horizon (e.g., one 24 hour period).

Maximize:

$$\sum_{\mathcal{M}} \sum_{l \in L_m} \sum_{\mathcal{F}} x_{m^l f} \left(\text{risk}_f \cdot \text{importance}_f \right)$$

Subject to:

(No more than 1 marshal/flight)

$$\sum_{\mathcal{M}} \sum_{l \in L_m} x_{m^l f} \leq 1 \qquad \forall f \in \mathcal{F}$$

(Flight leg de-confliction constraints)

$$x_{m^{l2} f2} \text{t\_dep}_{f2} - x_{m^{l1} f1} \cdot \text{t\_arr}_{f1} < 0 \qquad \forall m \in \mathcal{M}; (l1, l2) \in L_m; (f1, f2) \in \mathcal{F}$$

$$x_{m^{l2} f2} \text{loc\_dep}_{f2} - x_{m^{l1} f1} \cdot \text{loc\_arr}_{f1} = 0 \qquad \forall m \in \mathcal{M}; (l1, l2) \in L_m; (f1, f2) \in \mathcal{F}$$

(Daily working hours)

$$\sum_{l \in L_m} \sum_{\mathcal{F}} x_{m^l f} (\text{t\_arr}_f - \text{t\_dep}_f) \leq \text{T}_{\text{workday}} \qquad \forall\, m \in \mathcal{M}$$

$$x_{m^l f} \in \{0,1\}$$

$x_{m^l f}$	decision to allocate leg l of marshal m to flight f
$\mathcal{F}$	set of all flights f
$\mathcal{M}$	set of all marshals m
L_m	set of flight legs l for marshal $m \in \mathcal{M}$
t_arr_f	arrival time of flight f
t_dep_f	departure time of flight f
loc_arr_f	arrival airport (location) of flight f
$\text{loc}_{\text{dep}_f}$	departure airport (location) of flight f
risk_f	contribution value associated with risk of flight f
importance_f	scaling factor associated with importance of flight f
$\text{T}_{\text{workday}}$	maximum time in marshal work day

REFERENCES

Airport Data & Contact Information. (2014, April 3). *Federal Aviation Admin*. Retrieved May 30, 2013, from http://www.faa.gov/airports/airport_safety/airportdata_5010/

Apostolakis, G. E., & Lemon, D. M. (2005). A screening methodology for the identification and ranking of infrastructure vulnerabilities due to terrorism. *Risk Analysis*, *25*(2), 361–376.

ASPM Airports - ASPMHelp. (n.d.). Retrieved August 1, 2013, from http://aspmhelp.faa.gov/index.php/ASPM_Airports

Balakrishna, P. (2009). *Scalable approximate dynamic programming models with applications in air transportation*. George Mason University. Retrieved from http://digilib.gmu.edu:8080/jspui/handle/1920/5619

Barnett, A. (2004). CAPPS II: The foundation of aviation security? *Risk Analysis*, *24*(4), 909–916.

Barnhart, C., Belobaba, P., & Odoni, A. R. (2003). Applications of operations research in the air transport industry. *Transportation Science*, *37*(4), 368–391.

Barnhart, C., Cohn, A. M., Johnson, E. L., Klabjan, D., Nemhauser, G. L., & Vance, P. H. (2003). Airline crew scheduling. In *Handbook of transportation science* (pp. 517–560). Springer. Retrieved from http://link.springer.com/chapter/10.1007/0-306-48058-1_14

Bellman, R. (2003). *Dynamic Programming*. Mineola, NY: Dover Publications.

Bier, V., Azaiez, M. N., & Cox, L. (2009). Why Both Game Theory and Reliability Theory Are Important In Defending Infrastructure Against Intelligent Attacks. *Published Articles & Papers*. Retrieved from http://research.create.usc.edu/published_papers/89

Bier, V. M., & Azaiez, M. N. (2009). *Game Theoretic Risk Analysis of Security Threats*. Springer.

Biles, C. (2013, December 16). 5 Ways Federal Air Marshals Have Changed Since 9/11. *Business Insider*. Retrieved April 12, 2014, from http://www.businessinsider.com/5-ways-federal-air-marshals-have-changed-since-911-2013-12

Castelli, M., Meier, T., Morris, M., Philie, M., & Kwinn, M. (2013). The Federal Air Marshal Service: Using Value Focused Thinking To optimize field office allocations. In *2013 IEEE International* (pp. 809–811). West Point, NY: United States Military Academy.

Clarke, M., & Smith, B. (2004). Impact of Operations Research on the Evolution of the Airline Industry. *Journal of Aircraft*, *41*(1), 62–72.

De Rugy, V. (2005). What does homeland security spending buy? *American Enterprise Institute for Public Policy Research*. Retrieved from http://www.aei.org /files/2005/04/01/20050408_wp107.pdf

Eric V. Denardo. (1982). *Dynamic programming : models and applications*. Englewood Cliffs, NJ: Prentice-Hall.

FAA Operations & Performance Data. (2014). *Federal Aviation Administration*. Retrieved May 30, 2013, from https://aspm.faa.gov/

Federal Air Marshals. (2013). *Transportation Security Administration*. Retrieved April 16, 2014, from http://www.tsa.gov/about-tsa/federal-air-marshals

Gaibulloev, K., Sandler, T., & Sul, D. (2013). Common drivers of transnational terrorism: Principal component analysis. Retrieved from http://research.create.usc.edu /published_papers/144/

George, A., Powell, W. B., Kulkarni, S. R., & Mahadevan, S. (2008). Value function approximation using multiple aggregation for multiattribute resource management. *Journal of Machine Learning Research*, *9*, 2079–2111.

Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer Academic Publishers.

Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., & Ordóñez, F. (2010). Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshal Service. *Interfaces*, *40*(4), 267–290. doi:10.1287/inte.1100.0505

John Moteff. (1953). *Risk Management and Critical Infrastructure Protection: Assessing, Integrating, and Managing Threats, Vulnerabilities, and Consequences*. Harper & Brothers.

List of hub airports. (2014, April 16). *Wikipedia, the free encyclopedia*. Retrieved May 30, 2014, from http://en.wikipedia.org/wiki/List_of_hub_airports

Lord, S. (2010). *Federal Air Marshal Service: Actions Taken to Fulfill Core Mission and Address Workforce Issues: Congressional Testimony*. DIANE Publishing. Retrieved from http://www.gao.gov/assets/130/123101.pdf

Mahadevan, S., & Maggioni, M. (2006). Value function approximation with diffusion wavelets and Laplacian eigenfunctions. *Advances in Neural Information Processing Systems*, *18*, 843.

McLay, L. A., Lee, A. J., & Jacobson, S. H. (2010). Risk-Based Policies for Airport Security Checkpoint Screening. *Transportation Science*, *44*(3), 333–349. doi:10.1287/trsc.1090.0308

Nikoofal, M. E., & Zhuang, J. (2012). Robust Allocation of a Defensive Budget Considering an Attacker's Private Information. *Risk Analysis*, *32*(5), 930–943. doi:10.1111/j.1539-6924.2011.01702.x

Petrik, M. (2007). An Analysis of Laplacian Methods for Value Function Approximation in MDPs. In *IJCAI* (pp. 2574–2579). Retrieved from http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-414.pdf

Powell, W. B., & Wiley InterScience (Online service). (2011). *Approximate dynamic programming solving the curses of dimensionality*. Hoboken, N.J.: Wiley. Retrieved from http://site.ebrary.com/id/10501323

Regions and Aeronautical Center Operations. (2012, August 3). *Fed*. Retrieved May 30, 2013, from https://www.faa.gov/about/office_org/headquarters_offices/arc/

Risk-Based Allocation of Resources to Counter Terrorism - CREATE. (n.d.). Retrieved July 31, 2013, from http://create.usc.edu/2009/05/kleinmuntz_willis_2009_-_risk-.html

Secure Flight Program. (2014, March 19). *Transportation Security Administration*. Retrieved April 17, 2014, from http://www.tsa.gov/stakeholders/secure-flight-program

Simão, H. P., Day, J., George, A. P., Gifford, T., Nienow, J., & Powell, W. B. (2008). An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. *Transportation Science*, *43*(2), 178–197. doi:10.1287/trsc.1080.0238

Spivey, M. Z., & Powell, W. B. (2004). The dynamic assignment problem. *Transportation Science*, *38*(4), 399–419.

Strasser, S., & Whitney, C. R. (2004). *The 9/11 Investigations: Staff Reports of the 9/11 Commission: Excerpts from the House-Senate Joint Inquiry Report on 9/11: Testimony from Fourteen Key Witnesses, Including Richard Clarke, George Tenet, and Condoleezza Rice*. New York: Public Affairs.

Tsai, J., Kiekintveld, C., Ordonez, F., Tambe, M., & Rathi, S. (2009). IRIS-a tool for strategic security allocation in transportation networks. Retrieved from http://research.create.usc.edu/nonpublished_reports/71/

USGPO. (2007). *The secretary's second-stage review: re-thinking the Department of Homeland Security's organization and policy direction* (Hearing No. 109-32). U.S. Congress.

Willis, H. H., Morral, A. R., Kelly, T. K., & Medby, J. J. (2005). *Estimating terrorism risk*. Santa Monica, CA: RAND Corporation.

Wright, P. D., Liberatore, M. J., & Nydick, R. L. (2006). A Survey of Operations Research Models and Applications in Homeland Security. *Interfaces*, *36*(6), 514–529. doi:10.1287/inte.1060.0253

BIOGRAPHY

Keith W. DeGregory was born in 1975. He grew up on Long Island, NY raised by his parents Gilbert and Loretta and living with his four siblings: Richard, Diane, David, and Kevin. He is an active duty Army officer of 16 years currently serving in the career field of Operations Research and Systems Analysis (ORSA). He has maintained a balance of operational experience and academia over the past 20 years. He graduated from the United States Military Academy with a bachelor of science in System Engineering in 1997 and was commissioned as a second lieutenant in the U.S. Army in the branch of Air Defense Artillery. He has served on three operational tours in three different countries supporting the war on terror. As a company commander he deployed his battery to Jordan in 2003. In 2005 he served on the staff of Multi-National Corps – Iraq as the operations officer for the Anti-Terrorism and Force Protection section. His last deployment (2010-11) was as the Chief of the Operations Research section for Combined Joint Task Force Paladin in Afghanistan, the counter-improvised explosive device command for the International Security Assistance Force Joint Command (IJC). He holds a Master of Arts in Management from Webster University (2002) and a Master of Science in Operations Research from Massachusetts Institute of Technology (2007). He has served as an instructor and assistant professor in the Department of Mathematical Sciences at West Point, NY (2007-10). While there he taught and served as course director for the core differential calculus program. Keith is married to the former Christine Schuberth of Smallwood, NY and has three children: Hannah, Daniel, and Benjamin.