Approximate Dynamic Program for Scheduling Federal Air Marshals in Real-Time

Keith W. DeGregory
Department of Mathematics
United States Military Academy, West Point, NY 10996

Rajesh Ganesan
Department of Systems Engineering and Operations Research
George Mason University, Fairfax, VA 22030

Abstract

An approximate dynamic program (ADP) is proposed to assist schedulers in dynamically allocating air marshals to optimize the coverage of perceived risk across the nearly 30,000 daily domestic and international flights. The Federal Air Marshal Service (FAMS) provides front-line national security by deploying undercover federal air marshals to commercial flights to detect, deter, and defeat hostile acts targeting the United States. This paper focuses on a method of scheduling a subset of FAMS serving in a reactionary role to maximize the coverage posed by a stochastic risk. The dynamic allocation of reactionary air marshals requires sequential decision making under uncertainty with limited lead time. The marshal allocation system is modeled as a Markov Decision Process. Value function approximation schemes are explored to mitigate scalability challenges by alleviating the need for explicit state value storage. The study demonstrates that air marshal allocation in near real-time is possible using an ADP to determine scheduling policy. Results demonstrate improved coverage of stochastic risk using ADP over the myopic approach (pre-scheduling) on a subset of the domestic flights over a ten day horizon.

Keywords

1. Introduction

This paper addresses national security by proposing an approach and method for dynamically allocating federal air marshals to passenger flights to protect against a stochastic risk. This will serve to improve national security by increasing protection to commercial aviation from potential hijackings or future attacks using passenger jets such as those experienced on 9/11. The mission of the Federal Air Marshal Service (FAMS) is to promote “confidence in the nation’s civil aviation system through the effective deployment of Federal Air Marshals (FAMs) to detect, deter, and defeat hostile acts targeting U.S. carriers, airports, passengers, and crews” [1]. This paper focuses on the “effective deployment” referenced in the mission statement.

The current number of FAMs and the existing deployment procedures in force are sensitive security information to which the author is not privy. The approach in this paper assumes that dynamically scheduling the entire set of FAMs on a daily basis would result in unsustainable operations (e.g., persistent unpredictability in work schedules leading to burn-out and high marshal turnover). As such this research proposes the creation of a quick reaction force (QRF) consisting of a small subset of marshals that are allocated to higher risk flights in real time. Marshals would rotate on to the QRF for a short duration (e.g., 7-10 days) followed by a longer stint of predictable scheduling (e.g., 3 months). While serving on the QRF, marshals are assigned to flights, within the hours leading up to departure, based upon a stochastic risk. We present a model and approximate dynamic program (ADP) to assist schedulers in allocating QRF marshals in real time in order to most effectively maximize coverage of the riskiest flights [2].
1.1 Literature Review
The commercial aviation industry has long attracted operations researchers due to its data-rich environment and wealth of opportunities to improve operations. The industry consists of complex and interconnected sub-systems, many of which are plagued with inefficiencies. Left unaddressed, such inefficiencies can contribute to the failure of major airlines. One area of this industry that is particularly popular amongst researchers is the crew scheduling problem. A common theme is the generation of subsets of feasible flight-crew pairings in which the sequence of flight legs are grouped such that flight pairings originate and terminate at the same location and satisfy the many constraints such as consecutive hours on duty [3]. At its core, the crew scheduling problem is a set partitioning and set covering problem where the combinations of all possible pairings are endless. FAMS schedulers face a similar challenge to that of the crew scheduling problem with one main difference: air marshals are not required to cover every flight nor could they due to their limited numbers. This research will address how to prioritize the allocation of marshals in light of this fact.

Other literature focused specifically on the FAMS and security of commercial aviation exists, but it is less prevalent than research focused on economic factors and outcomes. One such paper discussed the scheduling of air marshals as a Stackleberg game and introduced software assistants to employ randomization in the scheduling of marshals. The Intelligent Randomization in Scheduling (IRIS) approach developed by the authors mitigates predictability in scheduling by avoiding the scheduling of marshals to only the highest tier of riskiest flights [4]. The research presented in this paper also sets out to avert unpredictability but in a different manner. The ADP approach reduces predictability but accomplishes this by delaying allocation of marshals until closer to scheduled departure times and selecting flights according to a stochastic risk that stems from the composition of the manifest. We stipulate that such an approach would require a passenger tracking and cueing system similar to the Computer-Assisted Passenger Prescreening System (CAPPS) II proposed after 9/11. The CAPPS II system was never implemented amidst privacy and civil-liberty concerns raised by such organizations as the American Civil Liberties Union [5].

In another article airport security is addressed via risk-based policies for checkpoint screening at airports [6]. The authors use a Markov Decision Process and ADP to compute the optimal policy for assigning security devices to passengers assessed at different risk classes upon check-in. This is a similar approach to that discussed in this research; however, the security asset in the FAMS problem are the actual marshals, which require dynamic repositioning in time and space, and the target for the asset is the aggregation of risk associated with an entire manifest and not individual passengers. We focus the model discussed in this paper to address this dynamic component and acknowledge the potential for a greater benefit if both approaches are employed synchronously.

1.2 Problem Statement
How can the proposed QRF (a small subset of the larger set of air marshals) be allocated to flights in such a way to maximize effective deployment of these marshals? Inherent in this problem is determining an appropriate metric for measuring effectiveness and determining how to attribute risk. This problem is characterized by sequential decision making under uncertainty in which decisions about allocating marshals to flights must be made sequentially and the uncertainty presents itself in the form of a stochastic risk to flights particularly arising from the composition of passenger manifests. This relies on the assumption that the greatest risk is posed by a terrorist who is able to circumvent all other levels of security and board a commercial flight posing as a normal passenger.

2. Research Approach
Dynamic programming (DP) is suited for problems of this nature but the approach does not scale to large problems. Thus we will apply a variant of DP, approximate dynamic programming, in which we generate an approximating function in lieu of explicitly storing the values associated with the states in the system. We model state transitions according to a Markov Decision Process. We adopt the notation and algorithms presented by Powell [7].

2.1 Markov Decision Process
First we model the system as a Markov Decision Process (MDP). In a standard Markov Chain we start with a directed graph \( G = (S, E) \) representing the system of possible states \( S \) and edges \( E \). Transitions from state \( s_i \) to \( s_j \) occur at uniform time steps according to probability \( p_{ij} \) and only depend on the current state, not the history of preceding transitions. Such a system exhibits the memoryless property. We adapt the Markov Chain by introducing actions or decisions that can be taken which can influence the transition probabilities. Furthermore, each decision results in a contribution of \( C(s_j | s_k, x_k) \) given we arrive at state \( s_j \) from state \( s_k \) under decision \( x_k \). The contribution can either be
a positive value (reward) or negative (penalty) for a given maximization problem. The goal of the MDP is to find the best policy \( X^n(S) \) to follow to produce the greatest long run expected contribution. The best policy in an MDP maximizes the long run contribution captured in the objective function of the infinite horizon problem (\( \gamma \) represents a discount factor which places emphasis on short term rewards):

\[
\max_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t=0}^{\infty} \gamma^t C_t(S_t^n, X_t^n(S_t)) \right\}.
\]

### 2.2 Approximate Dynamic Program
ADP utilizes Bellman’s optimality equations to recursively update and store state values within an approximating function \( \tilde{V}(S) \) [8]. This is accomplished through learning based simulation in which MDP occurs according to a two stage transition model \( M \) at each iteration \( n \) of the simulation. The first stage is a transition to a post-decision state (PDS) \( S^{x,n} \) according to the model \( S^{M,x}(S^n, x^n) \) and the second stage is a transition to the next pre-decision state \( S^{n+1} \) according to the model \( S^M(S^{x,n}, W^{n+1}) \) where \( W^{n+1} \) represents an exogenous stochastic process occurring between iteration \( n \) and \( n + 1 \). We incorporate Bellman’s optimality equations into the simulation at each iteration by solving the decomposed problem

\[
\hat{V}^n = \max_{x \in X^n} \left( C(S^n, x^n) + \gamma \tilde{V}^{n-1}(S^{M,x}(S^n, x^n)) \right).
\]

In this problem \( \hat{V}^n \) represents the estimated value under decision \( x \) which solves the maximization problem, where \( C(S^n, x^n) \) represents the contribution associated with making decision \( x^n \) from state \( S^n \) and \( \tilde{V}^{n-1}(S^{M,x}(S^n, x^n)) \) represents the stored value from the previous iteration of the PDS under decision \( x^n \). We experiment with techniques of aggregation and regression via diffusion wavelets (covered in greater detail in original dissertation [2]) for the FAMS problem. The next step within the simulation is to update the value of the PDS according to the linear combination

\[
\tilde{V}^n(S^{x,n}) = (1 - \alpha_{n-1}) \tilde{V}^{n-1}(S^{x,n}) + \alpha_{n-1} \hat{V}^n,
\]

where \( \alpha \) is a learning parameter taking on values between 0 and 1. The learning parameter is initialized at a value close to 1 and monotonically decreases over the course of the simulation. Careful consideration of decay scheme for \( \alpha \) must ensure \( \alpha \) does not reach zero prior to meeting convergence criteria. Next we transition according to \( \omega \in W \) to the next pre-decision state. The algorithm then checks if convergence criteria is met, if so the simulation ends and returns the optimal approximating function used for determining optimal policy. The complete algorithm:

1. Initialize \( \bar{V}^0, S^0, \) and \( \alpha^0 \) and set \( n = 1 \).
2. Solve \( \hat{V}^n = \max_{x \in X^n} \left( C(S^n, x^n) + \gamma \tilde{V}^{n-1}(S^{M,x}(S^n, x^n)) \right) \)
   and let \( x^n \) be the value of \( x \) that solves the maximization.
3. Find the post-decision state \( S^{x,n} = S^{M,x}(S^n, x^n) \).
4. Update the value approximation function for the post decision state
   \( \tilde{V}^n(S^{x,n}) = (1 - \alpha_{n-1}) \tilde{V}^{n-1}(S^{x,n}) + \alpha_{n-1} \hat{V}^n \).
5. Choose a sample path \( \omega^n \) and update \( \alpha^n \).
6. Find the next pre-decision state \( S^{n+1} = S^M(S^{x,n}, W(\omega^n)) \).
7. Check for convergence, if convergence criteria met, return final value approximation function, else go to Step 2.

Using an approximating value function makes possible the ability to solve large-scale dynamic problems that are often intractable using classical mathematical programming techniques such as linear or integer programming.

### 3. Model
The optimality equations and the algorithms presented in the last section address the science behind ADP. This section focuses on the art of ADP. Modeling the problem is as much art as it is science as every problem will have its own
unique attributes and structure. This section will describe how we model the FAMS operating environment, state variables, and decision space.

3.1 Data
The operating environment for federal air marshals intersects with that of the commercial air transportation industry. Components of the commercial air transportation relevant to this research consist of the physical airports, flight schedules, passenger manifests and passenger screening systems, security measures, and the actual flights. The following subsections briefly describes the various components included in the model.

3.1.1 Regions
We define region $r$ from the set of all regions $R$. This research utilizes three of the nine FAA administrative regions that divide the United States. The three regions include the Southern, Eastern, and Great Lakes encompassing a total of 21 states and the District of Washington. Regions contain the airports that are essential to modeling the system.

3.1.2 Airports
We define airport $a$ from the set of all airports $A$. Within the three regions we focus on the 20 airports that contain 90 or more flights per day. Eleven of these airports are considered major hubs allowing for many connection options; these hubs are an important component to the state space – to be discussed later. Airports serve as the arrival and departure locations for all flights in the system and staging grounds for all marshals.

3.1.3 Flights
We define flight $f$ from the set of all flights $F$. We restrict the model to flights that originate and terminate at one of the 20 airports making for a closed system. Each flight consists of defining attributes to include a unique identifier, departure airport, departure date time group, arrival airport, arrival date time group, risk level, and coverage indicator. The arrival and departure information used in the research is extracted from historical flight schedules [9]. The risk level of flights is discussed below and the binary coverage indicator is based upon whether a flight has a marshal allocated to it or not.

3.1.4 Marshals
We define marshal $m$ from the set of all marshals $M$. We restrict the model to a subset of $M$ that comprises the quick reaction force (QRF) referred to earlier. Marshal attributes include a unique identifier, domicile (home airport), duty tour start date, work day start time, daily flight allocations, projected location, and duty status. Marshals are constrained by a set of work rules. For example, they are limited by the number of consecutive hours they can be on duty and they must start and end their duty tour at their domiciles.

3.1.5 Risk
We recognize that assessed flight risk in practice would be generated from a combination of existing threat models and real-time cueing systems integrated into passenger screening systems and procedures. For this research we generate a stochastic model simulating the risk. We further institute the concept of risk thresholds which are based on tunable parameters that set by FAMS planners and leadership. In effect, the threshold would affect the sensitivity of the risk cueing system. For example, a larger threshold, say 3%, would mean that roughly 3% of the flights for that day would be classified as high-risk which means that more high-risk flights are likely to go uncovered if the number of QRF marshals remains relatively small. Setting the risk threshold higher has the effect of saturating the pool of riskiest flights making it harder to cover the riskiest of the high-risk flights. Conversely, reducing the risk threshold while holding the number of QRF marshals fixed enables focused coverage on the riskiest of high-risk flights.

3.2 Variables & Decision Space
This subsection describes the state variable and decision space of the model.

3.2.1 State Space
We define a state $s$ from the set of all states $S$. Since ADP relies on an MDP we must thoughtfully model the state space to ensure it encompasses all the information required to make decisions and transition to other states according to the model [7]. The state space for this problem must contain information on flights, marshals, and the threat in order to inform decisions. We examined a number of configurations for state space and ultimately selected the following: state $s$ is defined by three variables: a time block, a measure of marshal availability, and a measure of uncovered risk.
We define \( \tau \) as block of time from a discrete set \( T \) of time intervals that divide a 24 hour day. Inherent in this variable is information about density of flight departures and typical business hours. Marshal availability \( h_r \) is measured by maintaining the number of hubs by region that have at least one unallocated marshal currently located there or inbound in the immediate future. This variable relies on the assumption that hubs provide greater flexibility and connectivity to marshals in order to cover a high-risk flight or reposition throughout the work day. Lastly, uncovered high-risk flights by region \( U_r \) over a future time period for which marshals can still affect captures the demand for marshals. To reduce the state space this variable is further aggregated into three bins: bin 1 for no uncovered high-risk flights, bin 2 for between one and five uncovered high-risk flights, and bin 3 for six or more. This variable ignores uncovered high-risk flights in the immediate future (e.g., within 30 minutes of departure) as any unallocated marshal present at that departure location would have already been allocated. Thus the state \( s \) is compactly represented by the vector \( (\tau, h_r, U_r) \).

Considering the three regions in the reduced problem, the finite number of airport hubs, and use of bins for the measure of risk results in a system with 7,776 states for the case study. The relatively small state space will allow for explicit storage of all state values making it possible to measure the performance of state value approximation techniques against a lookup table of exact state values. For the larger problem spanning all nine US regions, including international flights, and/or expansion of state space variables, the number of states grows exponentially leading to \( NP \)-hard problems thus requiring the use of value function approximation.

### 3.2.2 Decision Space

The decision space for this problem consists of decision \( d \) from the set of all decisions \( D \). Decisions are made on a per marshal basis so an actual policy consists of a set of decisions for all active marshals currently awaiting orders during any decision cycle. In the reduced problem there are up to 10 decisions possible for any marshal. These decisions include actions such as: activating a marshal to duty for the day, allocating a marshal to a medium or high-risk flight, relocating a marshal within current region or to another region (via a low-risk flight), aborting a marshal from a scheduled flight in order to allocate to a high-risk flight, or no assignment (marshal waits in place). Each decision has an associated contribution; for example, covering a high-risk flight would provide more value than a covering a medium-risk flight. Additionally, penalties are assessed at the end of each decision cycle after a policy (set of decisions for each marshal) is executed for each high-risk flight that departs with no marshal coverage.

### 4. Experimentation

The experiment sets out to test the performance of scheduling QRF marshals using the learnt decision policy from the ADP algorithm against two basic strategies. The first is the myopic strategy which is a greedy approach set out to maximize the immediate reward at each decision point ignoring long term implications of those decisions. This strategy represents the lower bound for performance in that we assume this is our worst case scenario. The second strategy is designed to mimic decisions of a hypothetical experienced scheduler possessing knowledge of perfect information regarding the risk (thereby removing the stochasticity). The purpose of this latter strategy is to set a pseudo-upper bound on model performance. The primary metric selected to measure model performance is the percent of covered high-risk flights. Each experiment entails running 25 simulations consisting of 10 day blocks of time. During the experiment we test the sensitivity of the number of marshals as well as the risk threshold. We also explored a variant of our model that pre-allocated marshals to flights according to risks known at the beginning of the day and then allowed deviation from this initial daily allocation through the use of the abort and reallocate decision. We found in all cases this pre-scheduling technique was superior and thus we adopted it as a permanent feature to the model.

### 5. Findings

There are two main findings from our research. First, we demonstrate that it is possible to effectively allocate air marshals in near real time through the use of an optimal decision policy generated from an approximating value function learnt through simulation employing an ADP algorithm. The value function approximation technique employed performed as well as explicitly storing all state values in a lookup table. This finding is significant because it will allow schedulers to allocate marshals against dynamically changing risks within a relatively short window of final boarding times. This would make for effective deployment of air marshals against commercial flights posing the highest risk as determined by the composition of the flight manifest.

The second finding is that the learnt decision policy outperforms the myopic strategy and in a sampling of non-trivial cases reduces the critical coverage gap which describes the number of high risk flights missed between the myopic
DeGregory, Ganesan

(lower-bound) and experienced schedule with perfect information (upper-bound). For example, in one experiment that takes the average of 25 simulation runs consisting of approximately 2,850 daily flights scheduled between 20 airports over a 10 day period in which 20 QRF marshals are on call with the risk threshold set to 1%, we observe a coverage of 57% of high-risk flights missed under the myopic strategy. Table 1 shows the summary results from a number of similar experiments that vary the risk threshold while holding the number of marshals constant. Note that risk thresholds of two or more depict trivial cases where the number of flights categorized as high-risk is large relative to the number of QRF marshals, making it relatively easy to allocate marshals to high-risk flights (e.g., small Marshal:Risk ratio).

Table 1 Summary of Findings (Simulations with 20 marshals and 2850 flights)

<table>
<thead>
<tr>
<th>Risk Threshold (% of flights)</th>
<th>Marshal:Risk Ratio</th>
<th>High-risk Flights</th>
<th>Average High Risk Flight Coverage</th>
<th>Average Critical Gap Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Myopic (lower bound)</td>
<td>ADP</td>
<td>Exp. Scheduler w/ Perfect Info (upper bound)</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>4: 3</td>
<td>14</td>
<td>53%</td>
<td>67%</td>
</tr>
<tr>
<td>1</td>
<td>3: 2</td>
<td>29</td>
<td>48%</td>
<td>62%</td>
</tr>
<tr>
<td>2</td>
<td>1: 3</td>
<td>57</td>
<td>37%</td>
<td>53%</td>
</tr>
<tr>
<td>3</td>
<td>2: 9</td>
<td>86</td>
<td>30%</td>
<td>42%</td>
</tr>
</tbody>
</table>

A similar analysis is conducted on varying the number of QRF marshals while holding the risk threshold fixed at 1% with encouraging results [2]. Computationally the reduced problem runs in under six hours before converging to an optimal policy function while the large scale problem will likely take days. However, the long simulation times do not impact the viability of this approach because this is only an upfront cost. So long as the reading of state space variables is automated during the implementation phase, the policy vector can be calculated effortlessly by evaluating the approximation function for the current state which should take mere seconds. Execution of policy now depends on the speed at which the orders to the marshals on the QRF can be issued, which could also be systematically generated and sent via text or email. From these findings we conclude that it is not only feasible to schedule marshals in real time according to a learnt value function stemming from ADP but it outperforms the myopic strategy.

Acknowledgements
Federal Air Marshal Service, Department of Homeland Security; Center for Army Analysis, Ft. Belvoir, VA; Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, particularly Andrew Loerch, Lance Sherry, and Alexander Brodsky (Computer Science).

References