3-2021

The Autonomous Attack Aviation Problem

John C. Goodwill

The Autonomous Attack Aviation Problem

THESIS

John C. Goodwill, MAJ, US Army

AFIT-ENS-MS-21-M-162

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Army, the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-MS-21-M-162

THE AUTONOMOUS ATTACK AVIATION PROBLEM

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

John C. Goodwill, MS

MAJ, US Army

March 25, 2021

AFIT-ENS-MS-21-M-162

THE AUTONOMOUS ATTACK AVIATION PROBLEM

THESIS

John C. Goodwill, MS
MAJ, US Army

Committee Membership:

Dr. Matthew J. Robbins
Chair

Capt Phillip R. Jenkins, PhD
Member

AFIT-ENS-MS-21-M-162

Abstract

An autonomous unmanned combat aerial vehicle (AUCAV) performing an air-to-ground attack mission must make sequential targeting and routing decisions under uncertainty. We formulate a Markov decision process model of this autonomous attack aviation problem (A3P) and solve it using an approximate dynamic programming (ADP) approach. We develop an approximate policy iteration algorithm that implements a least squares temporal difference learning mechanism to solve the A3P. Several novel, problem-dependent basis functions are developed and tested for application within the ADP algorithm. The ADP targeting and routing policy generated by our algorithm is compared to a benchmark policy, the DROP policy, which is determined by repeatedly solving a deterministic orienteering problem as the system evolves over time. Designed computational experiments involving several problem instances are conducted to compare the benchmark and ADP policies with respect to their quality of solution, computational efficiency, and robustness. Quality of solution results indicate the ADP policy is superior in 2 of 8 problem instances investigated – those instances with relatively less AUCAV fuel availability and a low target arrival rate – whereas the DROP policy is superior in the remaining 6 of 8 problem instances. The ADP policy outperforms the DROP policy with respect to computational efficiency in all 8 problem instances investigated. The DROP policy provides more robust results, with less observed variance.

Key Words: Markov decision process, approximate dynamic programming, reinforcement learning, artificial intelligence, autonomous attack aviation, targeting, deep attack

To the less than 1% who choose to risk their lives for the other 99%,

To those who gave their lives honorably that we may live ours honorably,

May we be worthy of your sacrifice. I dedicate this research to you.

Acknowledgements

I thank my thesis advisor, Dr. Matthew Robbins, and reader, Dr. Phillip Jenkins for supporting me with this thesis. This work would not exist without my loving family and God's grace. Thank you.

John C. Goodwill

Table of Contents

Page

Page

List of Figures

List of Tables

THE AUTONOMOUS ATTACK AVIATION PROBLEM

I. Introduction

The purpose of the United States (US) military is to deter and win armed conflict. This raison d'etre requires the ability to destroy enemy forces and compel desired behavior. The defeat of adversary ground forces is a desired outcome of major combat operations, and military planners often task attack aviation assets to deliver lethal air-to-ground effects during such operations. The tasking of attack aviation assets is managed by a current operations targeting cell. This cell continuously monitors and adjusts current operations as new information becomes available. Routing attack aviation assets and delivering munitions requires deliberate management and oversight. The goal of attack aviation in high tempo combat operations is to deliver lethal effects as effectively and efficiently as possible, achieving the desired end state as envisioned by the commander. As with civilian-oriented service operations management, military service operations management, including employment of attack aviation assets, can greatly benefit from the application of operations research and artificial intelligence methods to gain and maintain a competitive advantage.

Contemporary military leaders agree that future combat operations will occur within complex and uncertain multidomain environments and that autonomous weapon employment is a critical component for achieving success in such operations (Mattis, 2018; Trends, 2017; Pellerin, 2015). In particular, the development of small, autonomous, and inexpensive aerial vehicles is of critical importance (Department of Defense, 2016; General Charles Q. Brown, 2020). It is essential for the US to quickly develop an effective and efficient autonomous unmanned combat air vehicle

(AUCAV) to gain a competitive military advantage over its adversaries (Department of Defense, 2018; Collins, 2020; Pasztor, 2021). Indeed, such adversaries seek to employ autonomous, lethal decision-making capabilities (Department of Defense, 2016; Department of the Army, 2011a; Naegele, 2020). The US military has allocated significant funds to prepare units for combat involving large Army formations operating against near-peer adversaries (Tressel, 2020).

A core capability of US air power is to conduct global strike missions, i.e., air-to-ground attacks, destroying enemy land forces anywhere on the planet using airborne assets. Two examples of air-to-ground attacks are close air support (CAS) and interdiction. CAS implies the presence of a nearby friendly ground force that requires coordination with the friendly air forces prior to conducting the air-to-ground attack. Interdiction does not require prior coordination with friendly ground forces. The purpose of joint interdiction operations is to "prevent adversaries from employing surface-based weaponry and reinforcing units at a time and place of their choosing" (Department of Defense, 2019c). US interdiction-capable forces must be able to employ lethal and non-lethal effects. Air interdiction is the ability for air forces to divert, disrupt, delay, or destroy an enemy's surface military assets (Department of Defense, 2020). A destroyed unit is physically rendered combat-ineffective until reconstituted (Department of the Army, 2018). This research models and analyzes the lethal effects of air interdiction. The process by which enemy assets are prioritized for attack is referred to as targeting.

Target selection and prosecution are critical processes that fuel interdiction. A target is anything that performs a function for an adversary (Department of Defense, 2020). Targeting is the iterative process that assigns friendly assets to a prioritized target list (Department of the Army, 2015). Selected indirect-fire weapons, ground maneuver forces, and attack aviation assets deliver desired effects to adversary targets.

Targets are categorized into two types, deliberate and dynamic. Deliberate targets are known to exist *a priori* and are typically scheduled or queued for prosecution whereas dynamic targets are unplanned or unanticipated targets that are identified too late in the targeting process or are discovered in an unanticipated location (Department of the Army, 2015). Dynamic targets are not scheduled for prosecution. Targets are arranged into a prioritized high-payoff target list (HPTL) that is generated by military planners prior to the dispatching of assets. With respect to attack aviation, dispatching includes a path assignment that starts at a point of departure following a predetermined path leading to a target area of interest (TAI) or kill box to acquire and engage high payoff targets (HPTs). In addition to TAIs, named areas of interest (NAIs) are identified by military planners as areas wherein enemy activities are anticipated, and the confirmation or denial of enemy activity in an NAI drives future military planning and targeting (Department of the Army, 2015).

Military operations are geographically segregated into deep, close, and support operations (Department of the Army, 2016). At the US Army division level, close and support areas are distinct, and responsibility for them is delegated to a subordinate unit. Deep operations are those military activities that occur within a unit's assigned area but not delegated to a subordinate unit. For the US military, deep operations normally fall under the purview of the division level. The division conducts deep operations with long range strategic assets to shape an enemy force before it becomes engaged with a friendly subordinate unit, such as a brigade combat team (BCT) (Department of the Army, 2014). The Division leverages missiles, artillery, and attack aviation assets to accomplish deep operations. An important concept to highlight with deep operations is that air-to-ground fratricide is extremely unlikely when physical distance to the closest friendly ground unit generally exceeds 10 kilometers.

In this thesis we consider the autonomous attack aviation problem (A3P) wherein a division-level current operations team must determine high-quality assignment and routing policies for attack aviation assets performing a deep air-interdiction attack mission. The mission objective is to engage a large-scale, near-peer mounted enemy ground force by destroying deliberate and dynamic HPTs through management of an HPTL while also conducting reconnaissance for targets via visitation of NAIs. Enemy ground targets, with their inherent, type-based priority, can be viewed as requests for service much like how profit-seeking companies view customers and their assessed demands for service. AUCAVs represent service providers, satisfying requests for service by delivering lethal effects to adversary ground targets. High-quality task assignment and routing decisions are required to achieve effective and efficient servicing of targets.

This thesis formulates a Markov decision process (MDP) model of the A3P. The uncountable state and outcome space renders classical dynamic programming techniques intractable. Instead, the problem solution methodology utilizes artificial intelligence to find an approximate solution via approximate dynamic programming (ADP) and reinforcement learning methods. A representative example of AUCAVs supporting deep interdiction targeting operations for a US Army Division in a defense against a contemporary near-peer, hybrid threat adversary is considered (Department of the Army, 2011a). A designed experiment is used to analyze problem features and algorithmic features.

The remainder of this thesis is structured as follows. Chapter 2 explores the relevant topical and methodical literature pertinent to the A3P. Chapter 3 introduces specific quantitative and qualitative problem features requisite to describe the problem, describes the MDP model formulation, and outlines the ADP solution approach. Chapter 4 presents computational results, analysis, and insights. Chapter 5 concludes the thesis and suggests future areas of research.

II. Literature Review

This section reviews the relevant literature, providing context for formulating the autonomous attack aviation problem (A3P). Four literature streams inform this research. First, recent formulations of the dynamic stochastic vehicle routing problem (DSVRP) provide a helpful mathematical modeling approach. Second, the literature related to the team orienteering problem with time windows (TOPTW) informs how aircraft choose some tasks over others. Third, literature related to Markov decision process (MDP) models utilizing ADP solution methodologies informs development of our own approach. We chose the term ADP in this thesis; however, the term reinforcement learning also applies. This stream is useful in that it not only succinctly models sequential decision-making under uncertainty, but also includes the powerful solution methodology of ADP to determine high-quality behavior policies. Finally, several works relevant to the construction of specific autonomous aircraft behaviors are discussed for breadth.

2.1 Dynamic Stochastic Vehicle Routing

The body of research related to dynamic stochastic problems was published within the last decade because computational power has only recently made the examination of large realistic problem instances possible. Surveys provide an excellent review of the background, taxonomy, and variants of the dynamic stochastic vehicle routing problem (DSVRP) (Pillac *et al.*, 2013; Psaraftis *et al.*, 2016; Ulmer, 2017). The DSVRP is relevant to this research because attack aviation assets (i.e., suppliers) operating in groups with limited payloads must efficiently visit enemy ground assets in unknown location (i.e., customers requiring lethal force). This section explores the background of DSVRPs, which informs the development of the A3P MDP model.

The vehicle routing problem (VRP) is a variant of the well-known traveling salesman problem (TSP). The TSP is a much studied problem in optimization wherein a traveler is compelled to visit an array of cities once using the shortest route possible and return to the starting city. The objective of the TSP is to find the route that minimizes the cost of traveling to each city once wherein cost is a function of distances among cities. The TSP is attributed to Flood (1956) and is the foundation to modeling many systems requiring service providers traveling to destinations in the form of a network. The VRP, originally referred to as the truck dispatching problem, is attributed to Dantzig & Ramser (1959). The VRP is a special case of the TSP wherein more than one traveler is required to visit a network of destinations only once. The starting node serves as the originating location for multiple travelers or vehicles with the same objective – to visit cities or locations only once and return to the starting node with the goal of minimizing travel cost. The starting node is commonly referred to as a depot. Vehicles are traditionally capacitated with a certain amount of supply, which they must use to satisfy the demand at each of the nodes along their assigned route.

The terms dynamic and stochastic refer to the level of uncertainty and change characterizing the VRP. A deterministic, static VRP is one in which the input parameters are known and stable, and no new actionable information is discovered as the system progresses (Ulmer, 2017). For example, if the location and demand for each node is known, the problem is deterministic. The problem is static if no new information is realized by the decision-making authority during the progression of the system through time. In a stochastic VRP the number of nodes, their locations, and demands are known but with uncertainty. Dynamic VRPs are those problems in which new information is realized during the progression of the vehicles traveling along their current assigned routes. For example, as a vehicle visits the second node along its assigned route of four nodes, the vehicle may discover that the demand is much lower than anticipated and potentially not spend as many supplies as previously expected. The structure of the DSVRP is conducive to modeling the A3P as an MDP. In contrast, the more common mixed integer program (MIP) formulation approach does not support the flexibility and powerful solution methodologies required to find a high-quality policy intended to inform sequential decision-making under uncertainty.

The purpose of a mathematical formulation of a real world problem is to summarize and record the salient features of the problem, to understand the problem, and apply a solution methodology. The MDP formulation of the DSVRP provides a unique linkage between applications and powerful solution methodologies (Ulmer *et al.*, 2017). Ulmer *et al.* (2020) provides a novel route-based MDP model that allows the decision-maker to not only integrate problem features into the model, but also facilitate the application of powerful solution methods such as ADP.

A route is a path through a set of realized service requests (Gendreau *et al.*, 1999). The route-based MDP is characterized by an action space comprised of a set of routes as opposed to a set of next nodes to visit. This inclusion of an entire route into an action space allows a decision-making authority to include deeper intuition in pursuit of a solution by allowing possible future rewards to impact the decision at the current time. The features introduced by Ulmer *et al.* (2020) are useful for modeling decisions in the A3P because it is prudent to consider the possibility of destroying yet-to-arrive, distant, high-value targets while making the decision to destroy closer targets in time and space. The DSVRP provides key components to the formulation of the deep A3P by effectively modeling relevant features and setting conditions to apply ADP solution methodologies.

2.2 Stochastic Dynamic Task-Resource Allocation Problem

In addition to vehicle routing, the stochastic dynamic task-resource allocation problem (SDTRAP) is relevant to the A3P because a set of AUCAVs must be assigned to prosecute a set of tasks in the form of enemy targets taking into consideration a finite set of weaponry (Gülpınar *et al.*, 2018). Gülpınar *et al.* (2018) present an MDP formulation as well as ADP solution methodologies to solve the SDTRAP. In the case of unmanned vehicles, the decision to task AUCAVs must be made jointly with resource allocation decisions (De Weerdt & Clement, 2009). It is prudent to assume that the weaponry of an individual attack aircraft is used at different rates when satisfying asynchronous demand requests.

The SDTRAP has successfully modeled complex and critical problems to include the transportation of a fleet of nationwide delivery vehicles, the management of investment within assets across a portfolio, and the management of blood resources across several healthcare provider systems (Powell, 2011). In consideration of the uncertainty of demand arrival and quantities of blood, complicated by the unique nature of constrained transmission of blood by blood type, hospitals must find a method to steward the precious blood resource to maximize health. Whether the resource is trucks, blood, dollars, or AUCAV resources, the management of limited resources to service uncertain and dynamic demand is a challenging and complex problem.

Elements from the DSVRP and SDTRAP are useful in structuring the A3P. Although no previous author has researched the A3P specifically, our problem formulation and solution methodology are built upon the critical works contributed by those researchers who have published in this methodological area.

2.3 Orienteering Problem

The orienteering problem (OP) is also a variant of the TSP albeit with two main differences. First, the requirement to visit all locations within a given set of service requests is relaxed. Second, the objective of minimizing cost is replaced with the objective of maximizing rewards gained by visiting a subset of the nodes within a set amount of time. The OP, originally introduced by Chao *et al.* (1996), is a problem wherein a traveler must start at a single location, visit as many locations as possible to satisfy demand, collect rewards, and return to their original position within a set amount of time. Similar to the VRP, the OP has several variants with unique features and characteristics. An interested reader may review a survey to gain an understanding of history, taxonomy, and applications of OPs (Vansteenwegen *et al.*, 2011).

The OP is relevant to the A3P because it is reasonable to expect that the objective of a team of AUCAVs is not to minimize cost. Rather, it is to maximize destruction of enemy targets. The OP clearly characterizes and mathematically models this important problem feature of the A3P. Prize collecting is the behavior of maximizing benefits when locations are visited (Balas, 1989). An OP variant conducive to the modeling of the A3P is the team orienteering problem with time windows (TOPTW) (Kantor & Rosenwein, 1992). This problem is appropriate because attack aviation missions prosecute targets with a small team of aircraft as opposed to a single aircraft operating independently. The time windows feature is useful to simulate the nature of targets only permitting servicing within a finite amount of time to earn rewards. An example of this is an enemy realizing they have been detected and move locations, or a target looses relevance before it is serviced.

Problems modeled as VRPs and OPs normally utilize an MIP formulation with a heuristic solution methodology. An effective model and solution approach for the A3P is an MDP model with an ADP solution methodology. The dynamic programming foundations found in ADP accommodate an uncountable state and/or action space while still providing a decision-making authority with a set of high-quality decision rules in the form of a policy.

2.4 Approximate Dynamic Programming

The nature of the A3P requires a decision-making authority to make sequential decisions under uncertainty with the information currently available. This makes dynamic programming the best suited modeling and solution methodology for modeling and solving the A3P. Several recent papers utilize ADP techniques to examine medical evacuation (MEDEVAC) dispatching problems, which exhibit similar characteristics as the A3P (Rettke *et al.*, 2016; Jenkins *et al.*, 2021). In Jenkins *et al.* (2021), air MEDEVAC assets are dispatched to casualty collection points predicated on uncertain arrival of casualties and types of casualties. The decision-making authority must decide when to dispatch MEDEVAC helicopters performing as service providers given an array of casualties or service requests. The objective of the MEDEVAC dispatch system is to evacuate combat casualties from casualty collection points to medical treatment facilities as effectively and efficiently as possible with the aim of preserving human life. The system is influenced by exogenous information generating uncertainty, which influences the sequential decision-making. In addition, the state and action spaces are too large for classical optimization techniques. The authors leverage ADP to simulate forward in time to generate sufficient estimates at the current time for the decision-making authority to take its best action. In the field of computer science, such methods are referred to as (model-based) reinforcement learning.

Other dynamic and stochastic problems that are effectively modeled via MDP and utilize ADP solution methodologies include shortest path problems, continuous budgeting problems, asset acquisition, asset liquidation, and dynamic resource allocation problems (Powell, 2011).

2.5 Unmanned Aerial Vehicle Behavior

Research related to military unmanned aerial vehicle (UAV) behavioral modeling is diverse with each piece of literature investigating a unique feature related to the A3P. Many researchers have investigated niche aspects of UAV behavior; however, less research exists regarding the blending of the multiple AUCAV behaviors required for realistic cooperative behavior in a deep air-interdiction scenario. For example, AUCAVs performing air-to-ground attacks have been investigated but only for one AUCAV. Other models involved multiple UAVs but only for surveillance missions. The remainder of this chapter is a survey of the salient features available in literature that are related to our problem. As outlined in Sutton & Barto (2018), specific feature research is critical for algorithm developers to take advantage of a problem's structure during modeling and solution development. We seek to study the specific nature of the A3P to enable precise modeling and useful solution development. A3P relevant features included in this literature review are: Route Planning, Task Assignment, Multi-UAV Task Cooperation, and Moving Targets.

2.5.1 Route Planning and Task Assignment

Route planning determines the movement path of an AUCAV or set of AUCAVs from their current position to their desired position. It may seem like a simple endeavor until accounting for flight dynamics. For example, an AUCAV has a minimum turning radius and the requirement to remain airborne. Much current research cites Dubins (1957) as a foundation to their analysis. Dubins (1957) proposes a simple procedure to plan paths considering constraints similar to those the physical world imposes on fixed wing aircraft like AUCAVs. Multiple methods have been used to determine the optimal path to include minimizing the total distance by way of a TSP (Medeiros & Urrutia, 2010), maximizing expected total discounted reward with a partially observable MDP (Widyotriatmo & Hong, 2008), and solving for the minimum time interval upon arrival (Li *et al.*, 2017). Many authors take unique approaches concerning how obstacles are considered in route planning.

Task assignment is the process of identifying service requests and matching servers or sets of servers to a service request or set of service requests. This can occur before, after, or potentially simultaneously with path planning. Task assignment must balance the reward from performing the task with the cost of traveling to the service site (Ning *et al.*, 2019). With respect to the A3P, a request for service is a ground target needing to be destroyed. Recent works on MEDEVAC task assignment by Rettke *et al.* (2016) and Jenkins *et al.* (2021) serve as a helpful baseline but do not focus on stochastic path planning or possible enemy components as are expected in near-peer direct action combat. Others consider not only the paths and motion of the AUCAV but also the ground service requests (Frew & Lawrence, 2005).

Some papers in the literature make a concerted effort to determine optimal route planning and task assignment simultaneously. Minimizing a cost function is a typical objective with many authors focusing on one AUCAV (Coutinho *et al.*, 2018; Zhang *et al.*, 2012) and others considering multiple AUCAVs (Zhu *et al.*, 2005). On occasion, authors embed optimization problems inside a broader optimization framework (Yang & Chakraborty, 2019). Each approach brings its own strengths; however, few have addressed the multiple moving AUCAVs serving multiple moving ground targets.

2.5.2 Muti-UAV Cooperation

Authors who have approached the multi-UAV problem normally offer a disclaimer that it is a difficult problem with many dynamic variables and uncertainties. A helpful theme in reviewed research includes assuming a heterogeneous makeup of AUCAVs to allow for tracking and attacking tasks to be assigned more appropriately (Valavanis & Vachtsevanos, 2015). In addition, simultaneous task execution is found to be a common obstacle to approach optimally (Shima & Rasmussen, 2009). Like all mathematical formulations of problems, the A3P must make prudent assumptions to generate a tractable mathematical model.

2.5.3 Moving Targets

A complicating problem feature within the A3P is considering multiple moving enemy ground targets whose route and speed is stochastic in nature. A solution approach identifies a likely meeting point to facilitate task assignment and route planning (Chen & Liu, 2019). Others have considered the use of a "dummy target" to facilitate solution procedure convergence in optimization (Jiang & Liang, 2018). In addition, a non-trivial problem feature is the number of AUCAVs relative to the number of moving ground targets.

III. Methodology

3.1 Problem Definition

In this section we describe the problem of developing a high-quality policy to govern the behavior of attack aviation assets performing a deep interdiction mission. We review the role of attack aviation within deep operations and describe the role of attack aviation in deliberate and dynamic targeting. We then introduce the strike coordination and reconnaissance (SCAR) mission. This mission provides a realistic context for the autonomous attack aviation problem (A3P).

3.1.1 Deep Operations

Attack aviation is a highly versatile military capability that contributes to the tactical, operational, and strategic level of war across the range of offensive, defensive, and stability operations. This research focuses on offensive, operational missions executed by US Army division-sized forces. The assets serving in the role of attack aviation in deep operations are of interest because these aviation assets are likely identified as the *main effort*. The main effort is a title given to the unit whose mission is so critical it is given priority for resources such as communications, artillery, and naval gun fire (Department of the Army, 2019b).

The nature of deep operations exhibits two critical attributes. First, deep operations exist beyond the forward line of troops (FLOT), which denotes the closest position of friendly ground forces. This extended distance relaxes the requirement for air assets to closely coordinate with ground forces, permitting attack aviation assets the flexibility to discover and prosecute targets unilaterally and quickly, with low risk of fratricide. The FLOT generally denotes the friendly boundary of the deep area. Figure 1 is an example graphical depiction of a US Army division conducting deep operations (Burket, 2019). The blue rectangle icons depict friendly units conducting operations from west to east, whereas the red diamonds denote enemy units. Black lines denote unit boundaries and control measures, which provide structure for planning and execution of operations. A single "X" denotes a Brigade Combat Team (BCT) and a double "XX" denotes the division headquarters. In this example, the FLOT is not fixed, but it resides within the Close Area, tethered to the three BCTs advancing from west to east within the Close Area. The coordinated fire line (CFL) depicted within the Deep Area is a line beyond which conventional forces may fire at any time within the intent of the CFL establishing headquarters (Department of Defense, 2019b). The BCTs are assigned missions focused on enemy units located within Objective A. Notice there are enemy forces within Objective B and beyond the CFL in the Deep Area where division assets, such as attack aviation, shape the ground fight for the maneuvering BCTs.

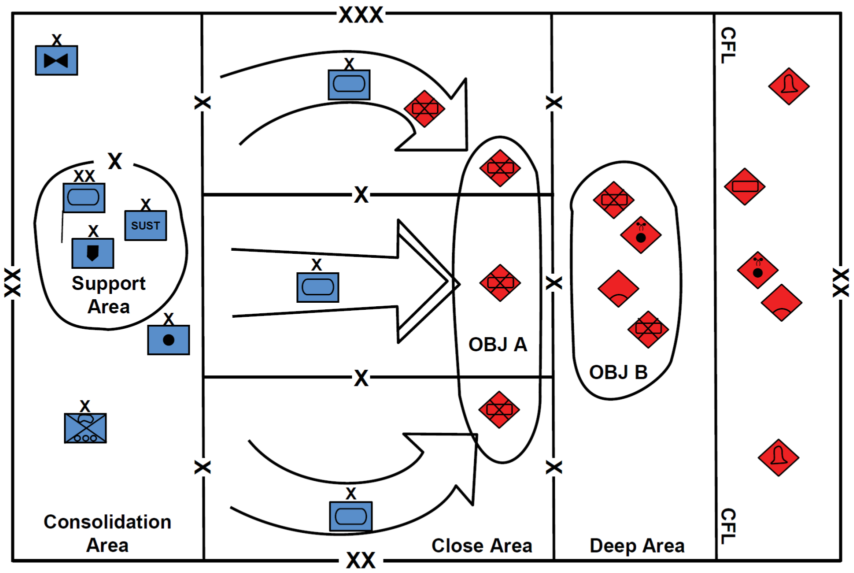


Figure 1. An Example Division Area of Operations (Burket, 2019)

Deep operations exist geographically in a linear or non-linear shape as shown in Figure 2. This highlights the non-trivial requirement to closely manage limited flight time and on-board resources to destroy known and prioritized enemy ground targets. If the mission's requirement exceeds the maximum weapon payload or the time one tank of fuel can provide, aviation assets must replenish their fuel and weapons at forward arming and refueling points (FARPs) to facilitate mission success. These FARPs are normally arrayed in fixed, secure locations to maintain sustained pressure on the enemy (Department of the Army, 2016). However, the US Air Force is currently testing mobile refueling of helicopters (King, 2020).

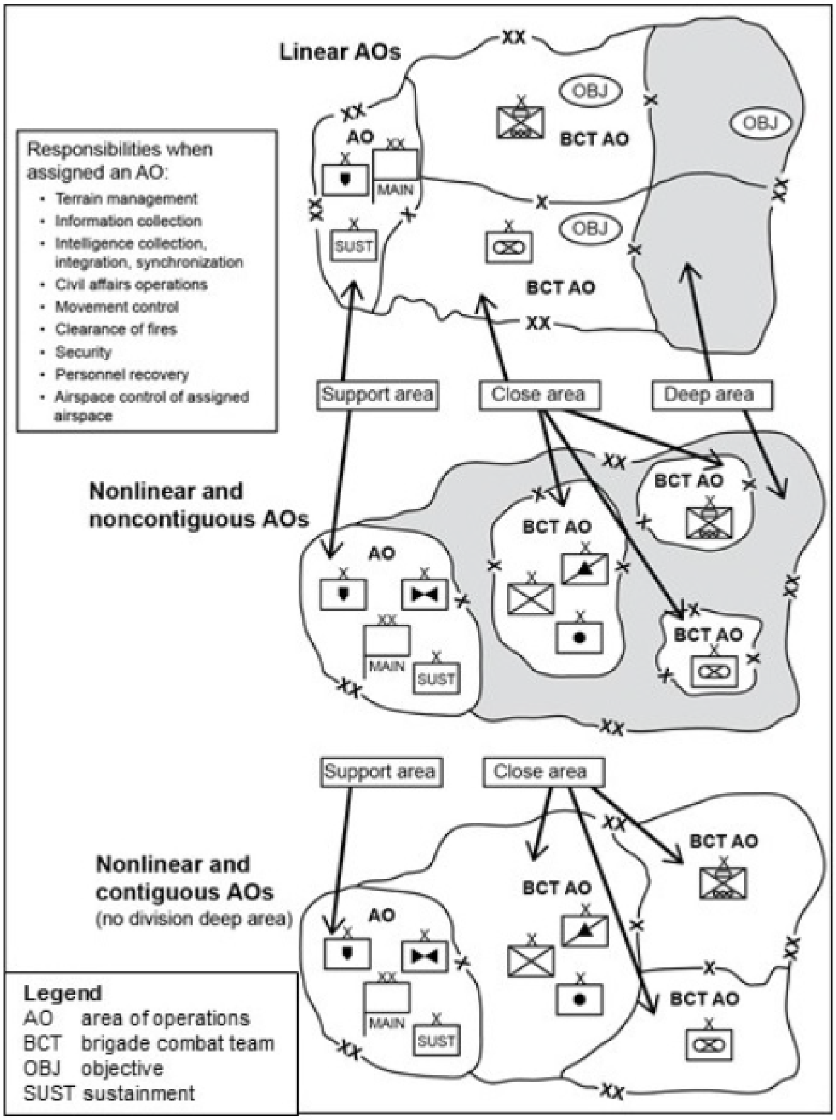


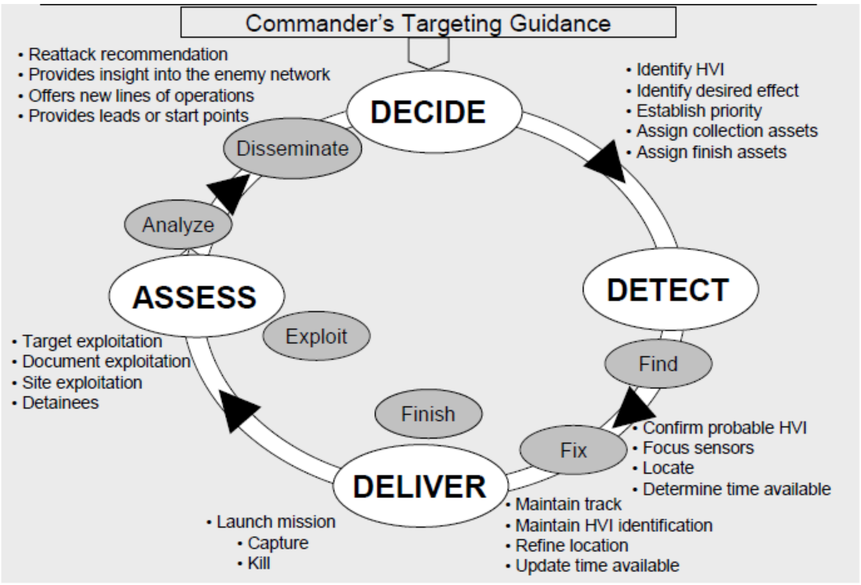
Figure 2. Example Geographic Framework (Department of the Army, 2016)

Attack aircraft do not always service ground targets with their own organic munitions. Indeed, it is often more conducive for an airborne asset to serve as a forward observer (FO) requesting friendly indirect fire to destroy a target it detects (Department of the Army, 2007). This function is helpful in two key ways: first, an FO who calls for indirect fire can attack without exposing their own presence; second, their attack does not expend any of their own finite organic ammunition. It is critical for attack aviation to be able to serve as an FO during a suppression of enemy air defense (SEAD) attack. Of course, aircraft prefer to attack enemy anti-aircraft assets as covertly as possible (Department of the Army, 2014). The DoD has shown recent interest in lethal munitions originating from various platforms while following the targeting guidance of aircraft (Freedberg, 2020; Gordon IV *et al.*, 2019). The US Army has recently demonstrated that their artillery achieves high accuracy over forty miles from the target (Judson, 2020a). An attack aircraft's capacitated fuel, and the AUCAVs role as an FO, inform development of our mathematical model.

3.1.2 Targeting

The success of deep operations depends upon the execution of effective targeting. Several teams share responsibility for a portion of the targeting process. These groups include the Deep Operations Coordination Cell (DOCC), the Joint Air Ground Integration Center (JAGIC), and the targeting working group. Each of these teams has a different role and responsibility, and the makeup of each team varies as a function of the division staff's mission and competencies. Ultimately, the responsibility for mission success lies with the Joint Force Commander (JFC). The targeting working group manages the targeting process (Department of the Army, 2015) within the military decision making process (MDMP). They are responsible for interpreting the JFC's guidance and generating the HPTL as part of the intelligence preparation of the battlefield (IPB).

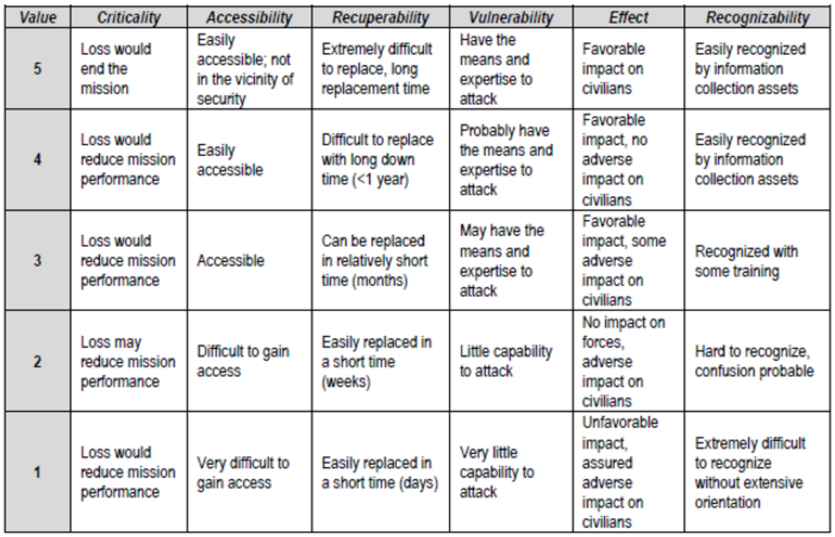
The targeting working group must effectively manage the execution of the HPTL to achieve mission success (Department of the Army, 2019a). The targeting process is summarized by the find, fix, finish, exploit, analyze, disseminate process (F3EAD) embedded in the decide, detect, deliver, assess (D3A) methodology. This process is used during both deliberate and dynamic targeting. Figure 3 illustrates the targeting process. This research focuses on lethal dynamic targeting.



Legend:
D3A – decide, detect, deliver, and assess F3EAD – find, fix, finish, exploit, analyze, and disseminate
HVI – high-value individual

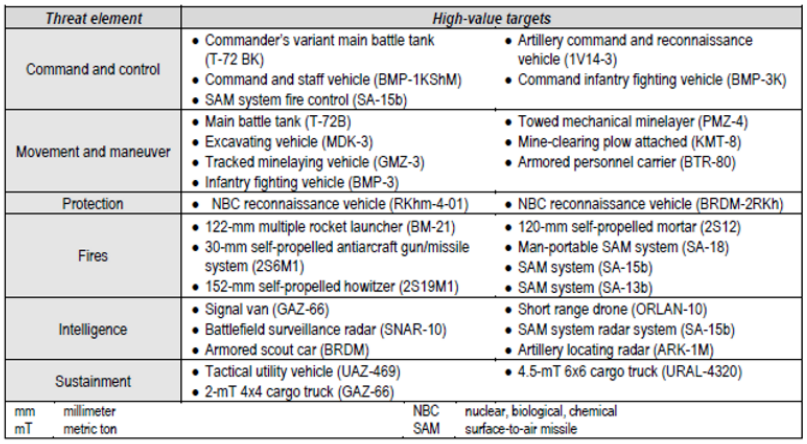
Figure 3. The Targeting Process (Department of the Army, 2015)

The lethal targeting of HPTs is intended to set conditions to achieve the purpose of a mission and to satisfy the JFC's intent. The value of individual HPTs is decided by the targeting working group and approved by the JFC before it is added to the HPTL. This process inherently takes deliberate analysis and requires time to complete. At the JFC's discretion, targeting may be generalized using a set of criteria to facilitate a faster target prosecution rate and increase the likelihood of mission success. A target value analysis tool that considers targets' criticality, accessibility, reputability, vulnerability, effect, and recognizability (CARVER) is used to identify and prioritize targets to efficiently assign attack resources (Department of the Army, 2019a). This tool, once understood by JFC staffs, can rapidly increase the rate of targets prosecuted, particularly in dynamic targeting – see Figure 4. In general, the lethal targeting of enemy high value assets includes anti-air capabilities, long range indirect fire assets, communications equipment, and all other enemy assets. An example HPTL is shown in Figure 5.



Value	*Criticality*	*Accessibility*	*Recuperability*	*Vulnerability*	*Effect*	*Recognizability*
5	Loss would end the mission	Easily accessible; not in the vicinity of security	Extremely difficult to replace, long replacement time	Have the means and expertise to attack	Favorable impact on civilians	Easily recognized by information collection assets
4	Loss would reduce mission performance	Easily accessible	Difficult to replace with long down time (<1 year)	Probably have the means and expertise to attack	Favorable impact, no adverse impact on civilians	Easily recognized by information collection assets
3	Loss would reduce mission performance	Accessible	Can be replaced in relatively short time (months)	May have the means and expertise to attack	Favorable impact, some adverse impact on civilians	Recognized with some training
2	Loss may reduce mission performance	Difficult to gain access	Easily replaced in a short time (weeks)	Little capability to attack	No impact on forces, adverse impact on civilians	Hard to recognize, confusion probable
1	Loss would reduce mission performance	Very difficult to gain access	Easily replaced in a short time (days)	Very little capability to attack	Unfavorable impact, assured adverse impact on civilians	Extremely difficult to recognize without extensive orientation

Figure 4. Example CARVER Matrix Tool (Department of the Army, 2019a)



Threat element	*High-value targets*	
Command and control	• Commander's variant main battle tank (T-72 BK) • Command and staff vehicle (BMP-1KShM) • SAM system fire control (SA-15b)	• Artillery command and reconnaissance vehicle (1V14-3) • Command infantry fighting vehicle (BMP-3K)
Movement and maneuver	• Main battle tank (T-72B) • Excavating vehicle (MDK-3) • Tracked minelaying vehicle (GMZ-3) • Infantry fighting vehicle (BMP-3)	• Towed mechanical minelayer (PMZ-4) • Mine-clearing plow attached (KMT-8) • Armored personnel carrier (BTR-80)
Protection	• NBC reconnaissance vehicle (RKhm-4-01)	• NBC reconnaissance vehicle (BRDM-2RKh)
Fires	• 122-mm multiple rocket launcher (BM-21) • 30-mm self-propelled antiarcraft gun/missile system (2S6M1) • 152-mm self-propelled howitzer (2S19M1)	• 120-mm self-propelled mortar (2S12) • Man-portable SAM system (SA-18) • SAM system (SA-15b) • SAM system (SA-13b)
Intelligence	• Signal van (GAZ-66) • Battlefield surveillance radar (SNAR-10) • Armored scout car (BRDM)	• Short range drone (ORLAN-10) • SAM system radar system (SA-15b) • Artillery locating radar (ARK-1M)
Sustainment	• Tactical utility vehicle (UAZ-469) • 2-mT 4x4 cargo truck (GAZ-66)	• 4.5-mT 6x6 cargo truck (URAL-4320)

mm	millimeter	NBC	nuclear, biological, chemical
mT	metric ton	SAM	surface-to-air missile

Figure 5. Example Target List (Department of the Army, 2019a)

During dynamic targeting, a sensor may detect a signal that is possibly a target but lacks sufficient fidelity to justify an attack. An emerging target is a detection that meets sufficient criteria to be acknowledged as a potential target (Department of the Army, 2015). Emerging targets are capricious, compelling the execution of a portion of the dynamic targeting process to gain more information. See the emerging target flow chart in Figure 6.

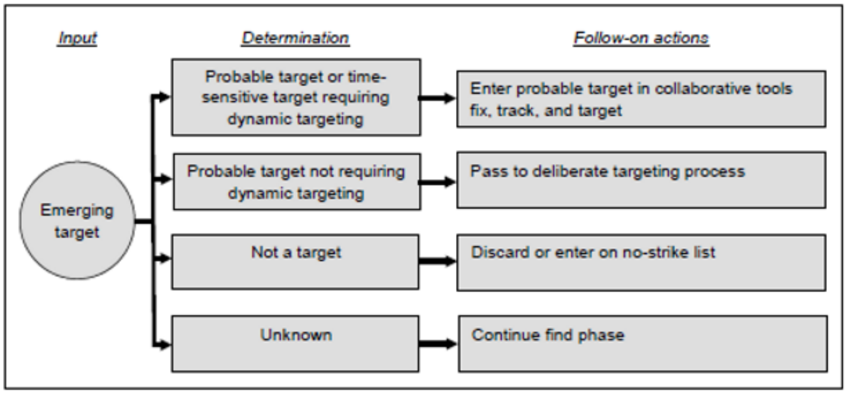

Input
Determination
Follow-on actions
Emerging target
Probable target or time-sensitive target requiring dynamic targeting
Enter probable target in collaborative tools fix, track, and target
Probable target not requiring dynamic targeting
Pass to deliberate targeting process
Not a target
Discard or enter on no-strike list
Unknown
Continue find phase

Figure 6. Find step determination and follow-on actions (Department of the Army, 2015)

Aviation assets performing a deep attack mission may encounter emerging targets en route to the prosecution of known targets. Some missions are launched based on this expectation. One mission type that showcases lethal dynamic targeting predicated on the expectation of encountering more targets is the SCAR mission. It is an intentional hybrid of both attack and reconnaissance (Department of Defense, 2019a).

3.1.3 SCAR Mission

The SCAR mission consists of Army aviation and joint assets. Such missions are flown to “detect and attack enemy ground targets, neutralize enemy air defenses, and provide battle damaged assessment (BDA)” (Department of the Army, 2016). BDA is the process of assessing the effect on the enemy after an attack. These three actions, detection, attacking, and BDA, are critical elements for our MDP model. In addition, the Joint Interdiction Manual highlights speed of intelligence and attacks as important (Department of Defense, 2019a). Referring to aviation assets in the deep attack, Army doctrine states “it is normally better to err on the side of speed, audacity, and momentum with the minimum mission essential information than waiting to gain complete situational understanding prior to conducting attacks” (Department of the Army, 2016; Judson, 2019a). The SCAR mission sets conditions for aviation assets to conduct rapid, lethal, dynamic targeting.

Static targeting is required to launch attack aviation assets in a SCAR mission. However, with new information discovered while executing the mission, dynamic targeting is likely the primary cause of fuel and weapon expenditure. The HPTL is not a static document because the dynamic targeting process is iterative. Not knowing your enemy’s composition and disposition with complete certainty is expected in armed conflict. The JFC’s staff initial expected geographic array of HPTs is referred to as a threat template. Figure 7 provides an illustration (Department of the Army, 2019a).

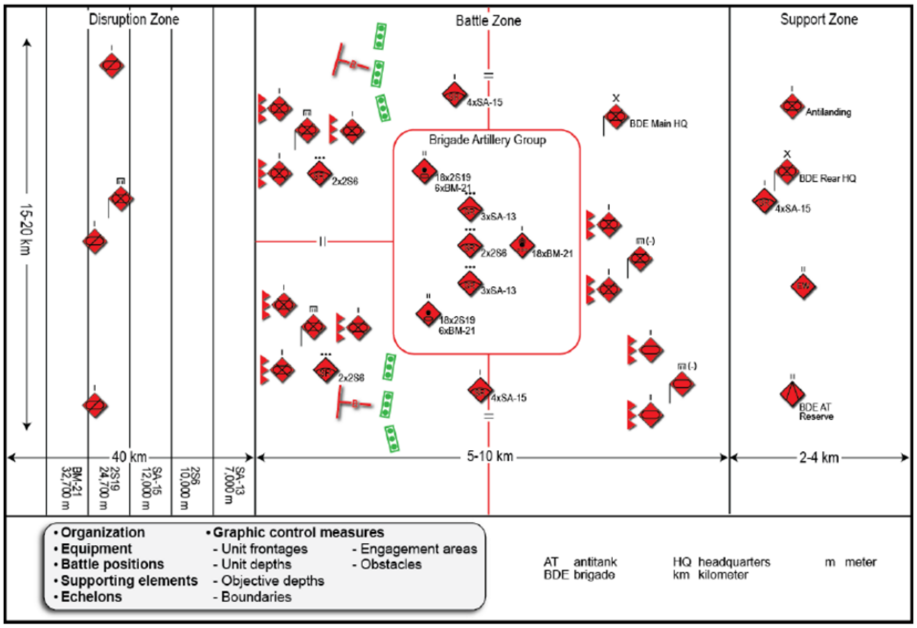


Figure 7. Threat Template Example (Department of the Army, 2019a)

Some targets are inherently static, like buildings or bridges. However, it is our assumption that all discovered targets will remain static indefinitely given the relative speed advantage the AUCAV has over ground vehicles. A non-static target disposition could manifest in the form of a target balking. Such a concept is worth examining in follow-on research. SCAR mission planners must prepare for and anticipate the expected, but unknown, to be successful. Moreover, speed of discovery, prosecution, and BDA are critical to the success of dynamic targeting.

3.1.4 The A3P

The A3P models a mission wherein an AUCAV tasked to the deep zone is assigned a SCAR mission given a fixed CARVER tool and initial HPTL. The HPTs are initially geographically arrayed at anticipated but unknown locations. This pre-mission intelligence assessment informs development of an initial launch route for the AUCAV. The AUCAV must discover, attack, and confirm destruction through BDA of as many HPTs as possible within a finite amount of time through dynamic targeting. The deep zone exit point is fixed and facilitates the AUCAV's decision to depart combat operations in route to the FARP. The deep targeting of enemy forces is operationally managed and synchronized by geographic control measures, including named areas of interest (NAIs) and target areas of interest (TAIs) (Department of Defense, 2019a). Human military planners geographically label expected clusters of anticipated enemy targets as TAIs to facilitate execution of operations. The NAI is an area wherein enemy activity is likely but uncertain, and the visitation to an NAI is expected to answer a JFC's critical information requirement (CCIR). The answers to CCIRs provide information regarding enemy composition, disposition, or terrain composition that are essential to future decisions the JFC expects to make during the course of a battle (Department of Defense, 2020).

In the following sections, we mathematically formulate the A3P as a Markov decision process (MDP) model, then introduce the approximate dynamic programming

(ADP) solution approach. A MDP model is utilized because the A3P requires sequential decision-making under uncertainty. The ADP solution approach is conducive to managing the uncountable state and action space within the MDP model of the A3P.

3.2 MDP Model

This section presents the mathematical formulation of the autonomous attack aviation problem (A3P) as a discounted, infinite-horizon MDP model. Model components include decision epochs, states, actions, transitions, and rewards (referred to as contributions). The objective of the model is to determine the best routing solution for an autonomous unmanned combat aerial vehicle (AUCAV) given a currently known geographic array of NAIs, the enemy ground targets as informed by the HPTL, and the status of friendly resources (i.e., playtime). The AUCAV is incentivized through the collection of rewards to choose the best action possible given current and expected future information. A strength of the MDP formulation over other modeling techniques is that the MDP model facilitates powerful modern approximate dynamic programming (ADP) solution methods, allowing A3P solution methods to anticipate and incorporate many realistic attributes of the real-world problem, thus facilitating high-quality solutions.

Table 1 includes definitions of key notation and acronyms to support reader comprehension of the remainder of the thesis.

3.2.1 Dynamic Targeting Process

The A3P consists of an initial threat template of known and suspected enemy positions produced from the IPB. The AUCAV is deployed on a SCAR mission and is repeatedly assigned the most effective and efficient route possible to prosecute known enemy targets, collecting the largest reward possible while playtime is

Table 1. Acronyms and Notation

Acronym	Description
AUCAV	Autonomous unmanned combat aerial vehicle
CCIR	Commander's critical information requirement
JFC	Joint forces commander (the responsible decision-maker)
IPB	Intelligence preparation of the battlefield
NAI	Named area of interest
TAI	Target area of interest
HPT	High payoff target
HVT	High value target (an asset adversary forces must have to accomplish their mission)
HPTL	High payoff target list (consists of HPTs and HVTs)
Notation	Description
π^{LSTD}	Least squares temporal difference policy
π^{DROP}	Deterministic repeating orienteering problem policy
S_t	state S at decision epoch t
ρ_t	playtime remaining at epoch t
θ	Coefficients (i.e, weights) of the approximate value function
α	Learning rate (i.e, smoothing) parameter
W_t	Exogenous information realized at epoch t

available. The model incorporates dynamic behavior of the A3P through numerous, sequential changes of the state of the system, through the realization of particular events. New information is discovered as the system progresses. Figure 8 depicts the inter-event process nested within the targeting process as experienced by the AUCAV.

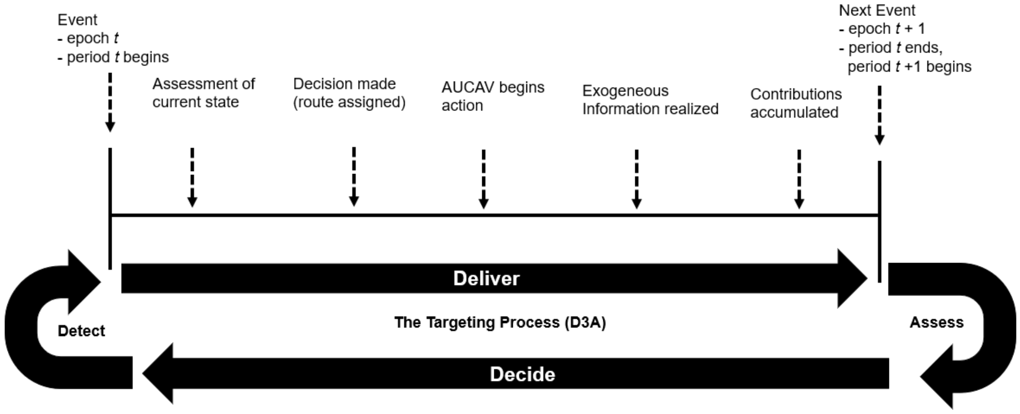

Event
- epoch t
- period t begins
Assessment of current state
Decision made (route assigned)
AUCAV begins action
Exogeneous Information realized
Contributions accumulated
Next Event
- epoch t + 1
- period t ends, period t +1 begins
Deliver
Detect
The Targeting Process (D3A)
Assess
Decide

Figure 8. The Inter-event Process

Events change the state of the system and require attendant decisions to be made at such decision epochs. The decision-making authority who controls AUCAV actions must evaluate the new state resulting from the event and make another decision to maximize rewards (i.e., contributions). The random, inter-event time between events is one decision period. Periods are not uniform in duration. A period may be near instantaneous or multiple minutes long. We use a Poisson process to model the stochastic arrival of new targets being scheduled for prosecution. When an emerging target is confirmed to be an HPT, it is scheduled and arrives to the HPTL from the targeting cycle (D3A). The inter-arrival time of targets to the HPTL follows an exponential distribution with rate λ. Once a target is placed on the HPTL, it is considered as a node requesting service. In the case of the A3P, this is an enemy ground target requesting destruction. Table 2 depicts the list of events that drive the evolution of the A3P system.

Table 2. A3P Event Types

1	Destruction of enemy HVT (service)
2	Destruction of enemy HPT (service)
3	AUCAV answers CCIR through visitation to NAI (service)
4	Emerging target approved and added to HPTL (arrival)

The A3P is an infinite horizon problem. Let $\mathcal{T} = \{0, 1, 2, ...\}$ denote the set of decision epochs over which decisions are made by the decision-making authority. Given the currently known HPTL, and the current position and status of the AUCAV, a decision-maker must determine the next target to visit to achieve the highest total contribution with the playtime remaining. We assume some deterministic behaviors in the A3P. The AUCAV's physical arrival to a target immediately results in confirmed destruction of the target through artillery or other lethal means. The AUCAV's physical arrival to an NAI immediately results in the answering of a CCIR associated with the NAI. Once a target or NAI is visited, it is not visited again. No reward is associated with TAI visitation. We also assume the AUCAV cannot be destroyed, and the AUCAV must return to the deep zone exit point before completely depleting the finite playtime allotted for the SCAR mission.

3.2.2 The State Variable

The state variable in Equation 1 consists of the minimal information necessary to generate an action, transition the system, and assess contributions (Powell, 2011). We denote the state of the A3P system at epoch t as

$$S_t = (A_t, R_t, N_t, \tau, e) \in \mathcal{S}, \tag{1}$$

wherein A_t is the AUCAV status tuple, R_t is the target service request status tuple, N_t is the NAI status tuple, τ is the current system time, e is the event type, and $\mathcal{S}$ is the set of all possible states. Let $\tau(S_t)$ and $e(S_t)$ denote the system time and the event type when the system is in state S_t at epoch t, respectively. We define the status tuple of the AUCAV at epoch t as

$$A_t = (\ell_t^A, \rho_t), \tag{2}$$

wherein $\ell_t^A \in \mathbb{R}^2$ denotes the two dimensional location of the AUCAV within the defined deep zone battle space and ρ_t denotes the playtime remaining. We assume the AUCAV travels at a constant speed, and fuel is depleted at a constant rate as the system time advances. This facilitates deterministic movement for the AUCAV.

The target service request status tuple $R_t = (R_{tr})_{r \in \mathcal{R}_t}$ contains the status of each target r in the set of known targets $\mathcal{R}_t$ as of epoch t. Let

$$R_{tr} = (\ell_{tr}^{R,HPT}, \xi_r), \tag{3}$$

wherein $\ell_{tr}^{R,HPT} \in \mathbb{R}^2$ denotes the two dimensional location of target request r within the defined deep zone battle space on the HPTL requesting destruction. The variable $\xi_r \in \{0, 1\}$ denotes the priority of the target wherein 0 denotes low priority and 1 denotes high priority. We assume once a target is added to the HPTL, its location and priority remain fixed unless visited by an AUCAV. When a target is destroyed, it is removed from the HPTL.

The NAI service request status tuple $N_t = (N_{tn})_{n \in \mathcal{N}_t}$ contains the status of each NAI n in the set of assigned NAIs $\mathcal{N}_t$ as of epoch t. Let

$$N_{tn} = (\ell_{tn}^{NAI}), \tag{4}$$

wherein $\ell_{tn}^{NAI} \in \mathbb{R}^2$ denotes the two dimensional location of NAI n within the defined deep zone battle space. When an NAI is visited, the CCIR is immediately answered and the visited NAI is removed from the set of active NAIs, $\mathcal{N}_t$.

We also define $O = (O_o)_{o \in O}$ as the set of TAI vertices whose locations, when connected denote a TAI polygonal area. $O_o \in \mathbb{R}^2$ denotes the two dimensional location of TAI vertices $o \in O$. The notation letter O is appropriate as a TAI is an operational graphic whose existence and placement is at the discretion of the controlling, human military forces. Note that the status of TAI vertices is fixed. The TAI vertices may not be visited as they do not demand service and the possible, but unlikely arrival of an AUCAV to a TAI vertex will not return a reward, as TAI visitation does not have its own intrinsic utility. TAIs are included, in part, to assist with development of anticipatory actions by the AUCAV, as is their purpose for human pilots. Their unchanging nature allows them to be excluded from the state variable.

3.2.3 Action Space

At each decision epoch t, the decision-making authority must choose which node to travel to next given the current state of the system. At the beginning of each decision epoch an AUCAV is located either (1) at the node which it selected as its next destination during the previous epoch, or (2) somewhere en route to its intended destination because its travel is interrupted by the stochastic arrival of a new target to the HPTL, initiating a new epoch. It is assumed that AUCAV travel time is deterministic. If a stochastic arrival does not occur, the AUCAV arrives precisely to its intended destination at a known arrival time. For example, suppose at time = 0, the AUCAV begins to travel to an NAI to answer a CCIR with a travel time of two minutes. Before the AUCAV arrives at the NAI, collecting the associated reward, suppose a new target arrives at time = 1 minute initiating a new decision epoch. If a stochastic arrival does not occur, the next decision epoch occurs at time = 2 minutes, upon the arrival of the AUCAV to its intended destination. It is possible the next resulting decision remains the same NAI chosen in the previous decision epoch. However, a re-evaluation of all feasible options occurs.

We denote $\mathcal{X}_{S_t}$ as the finite set of available actions given the current state S_t at epoch t. This comports with the traditional notation of x as the decision variable within the optimization discipline. The decision-making authority must choose one action x_t from set $\mathcal{X}_{S_t}$ for each epoch t. Each available action corresponds to the two dimensional location of the selected next node (i.e., NAI, target, or deep zone exit point). An AUCAV may choose any NAI or target on the HPTL that has not yet been visited, or the deep zone exit location. The set of available actions is

$$\mathcal{X}_{S_t} = \{(\ell_{tr}^{R,HPT})_{r\in\mathcal{R}_t},\ (\ell_{tn}^{NAI})_{n\in\mathcal{N}_t},\ \Omega\}, \quad \forall\ S_t \in \mathcal{S}, t \in \mathcal{T} \tag{5}$$

wherein Ω is the location of the single, fixed deep zone exit point. We assume the JFC always values the return of an AUCAV over the destruction of any amount of enemy targets or answered CCIRs. Hence, all feasible actions for a given epoch include only those destinations which the AUCAV has sufficient playtime to visit and return to the deep zone exit point. We also assume the AUCAV travels at a constant speed moving at a constant rate as the A3P system time progresses.

3.2.4 Transitions

The A3P system evolves over time changing the state of the system through an exogenous information process given the decision and stochastic information realization. The transition function describes how the system evolves. The stochastic nature of the system manifests in the uncertain arrival of targets to the HPTL according to a Poisson process as shown in Figure 8. As targets arrive, the system dynamically evolves to a new state and initiates a new decision epoch. Similarly, the arrival of an AUCAV to an NAI or target requesting service also causes an evolution of the system and initiates a new decision epoch. We let W_t denote the exogenous information realized at decision epoch t, and we denote the system model as

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}). \tag{6}$$

At each decision epoch t given the current state S_t, and decision x_t, the system model considers the resulting stochastic information W_{t+1} and evolves the system either to a new stochastic target arrival or the deterministic arrival of the AUCAV to the intended destination x_t. Recall the list of events in Table 2 which all result in a new decision epoch. In all cases, the system model S^M evolves the system from S_t to S_{t+1}.

3.2.5 Contributions

The contribution function motivates the desired behavior of each AUCAV in the A3P. It assigns a reward when a particular state is achieved by the system. In aggregation over the entire playtime, the contribution function quantifies the performance of a solution to the A3P, providing a total numerical value (i.e., total reward) after execution of all decisions and system evaluations for a given problem instance.

The A3P system collects rewards through the destruction of targets on the HPTL and answering the CCIRs through visitation of NAIs. Arrivals to any other location including TAI vertices, the deep zone exit location, and any random node on which an AUCAV may reside when interrupted by an arrival, does not return a reward. Rewards are collected immediately upon arrival to reward-bearing nodes. We denote the contribution function as follows

$$C(S_t) = \begin{cases} r^{HVT}, & \text{if } \ell_t^A = \ell_{tr}^{R,HPT}, \ \xi_r = 1 \\ r^{HPT}, & \text{if } \ell_t^A = \ell_{tr}^{R,HPT}, \ \xi_r = 0 \\ r^{CCIR}, & \text{if } \ell_t^A = \ell_{tn}^{NAI} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

These rewards induce the desired behavior while the autonomous aircraft is in communications with human decision-makers. We assume that if communications are ever severed between an AUCAV and the controlling human military forces, the AUCAV will return to the exit location and cease to conduct lethal operations in accordance with Department of Defense directive 3000.09: Autonomy in Weapon Systems Department of Defense (2017).

Some military practitioners may not be experienced in reinforcement learning and optimization. However, practitioners that utilize an autonomous capability similar to what we propose in the A3P may desire to be cognizant of this component of the mathematical model. The relative magnitude of the contributions in Equation 7 motivate the behavior of the aircraft and may help JFCs, and potentially human wingmen, understand why an AUCAV exudes the behavior that it does. The contribution function can be seen as a medium through which the responsible human military forces may choose to modify or tune AUCAV behavior to achieve the desired battlefield end state. The SCAR mission inherently values both reconnaissance (answering of CCIRs) and the destruction of enemy forces. In this event, a JFC may desire to have $r^{HPT} < r^{CCIR} < r^{HVT}$ or, if aggression is desired over reconnaissance, a JFC may desire $r^{CCIR} > r^{HVT} > r^{HPT}$ for that particular SCAR mission.

3.2.6 Objective Function and Optimality Equations

A policy is a decision rule mapping any given state to an action. Some decision rules are constructed to maximize the objective function via the Bellman Equation. We let $X^\pi(S_t)$ represent the decision function, which selects decision x_t given a specific state S_t, and policy π in decision epoch t. The optimal policy, π^*, is the policy that maximizes the expected total discounted reward (ETDR) of the MDP. The objective of our MDP model is

$$\max_{\pi \in \Pi} \mathbb{E}^\pi \Big[\sum_{t=1}^{\infty} \gamma^{\tau_t} C(S_t) \Big], \tag{8}$$

wherein $\gamma \in [0, 1)$ denotes the fixed rate discount factor, and τ_t denotes the system time at state S_t. Equation 9 provides the mechanism by which the optimal policy π^* is calculated, and is expressed as

$$V(S_t) = \max_{x_t \in \mathcal{X}_{S_t}} \Big(C(S_t) + \gamma^{(\hat{\tau}(S_{t+1}) - \tau_t)} \mathbb{E}\big[V(S_{t+1} \big| S_t, x_t \big] \Big), \tag{9}$$

wherein $\hat{\tau}(S_{t+1})$ is the time the system arrives to state S_{t+1}. The state space of the A3P is uncountable given that any target, NAI or an AUCAV could exist anywhere in $\mathbb{R}^2$. Given this condition, the optimal policy π^* for the A3P is unattainable using Equation 9. However, instead of relying on traditional optimization techniques, we leverage parametric value function approximation (VFA) through application of ADP. We compare our VFA approach to an available baseline policy through mean total reward earned through simulation. Given that the field of autonomous aircraft research is still young with many approaches to behavior control, we propose the deterministic repeated orienteering problem (DROP) solution approach as the baseline policy to support the effectiveness claims of our high-quality VFA approach. We choose the DROP baseline policy because the orienteering problem is well established in the literature and provides a known standard from which to measure performance. The mean total reward of the DROP and VFA approach are compared using a representative instance of the A3P.

3.3 ADP Solution Approach

ADP exists to overcome the curses of dimentionality that arise when attempting to solve large-scale dynamic programming problems. Described by Powell (2011), the curses of dimensionality refer to problems wherein the magnitude of dimensions are so high that an exact value of being in a certain state becomes impossible to calculate. Specifically, oversized dimensions manifest in the state space, action space, outcome space (resulting state), or any combination of the three. The A3P has an uncountable state space, and therefore an ADP approach is warranted to replace the exact value of being in any given state with an estimate of the value of being in a given state. This estimate is referred to as a VFA.

Two general approaches to approximating value functions exist. They are parametric and non-parametric models. Examples of non-parametric models include lookup tables such as aggregation schemes, Q-Learning, and neural network regression. Parametric models leverage regression methods, which presume the existing but unknown value function has structure. Such models must assume structure of the value function because they themselves have structure. To solve the A3P, we employ an approximate policy iteration (API) algorithmic strategy to construct a decision rule (policy) using a parametric VFA. We choose a parametric VFA since it permits applications to varying scales of a problem as well as facilitates the exploitation of structure within the model. An essential nuance of using parametric VFA is that its structure generally requires the generation of basis functions or features that are pieces of information provided by the state variable and are important when deciding which action to take.

3.3.1 Basis Functions

Effective basis function or feature construction is essential to generating accurate estimates of the value function. Basis functions serve as the independent variables in the continuous VFA. Let

$$\bar{V}(S_t|\theta) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t), \tag{10}$$

wherein $\bar{V}(S_t|\theta)$ denotes the approximation of the value function, such as Equation 9, given state S_t and value function approximation coefficients (or weights), $(\theta_f)_{f \in \mathcal{F}}$, and wherein ϕ_f is basis function and f is a feature in the set of features $\mathcal{F}$.

Our continuous VFA is a linear model. Therefore, it is essential we capture the most important features that aid decision-making. ADP generated policies are only as good as the accuracy of the value estimate. The closer the value estimate $\bar{V}(S_t)$ is to the true value $V(S_t)$, the closer the generated decision rule is to the optimal policy π^*. In the A3P, AUCAV behavior hinges on developing effective and efficient basis functions. We draw basis function insight from Rettke *et al.* (2016) and Jenkins *et al.* (2021) to inform formulation of our basis functions. Our basis functions must accurately and collectively capture the value of each future state to facilitate effective discrimination and selection of the highest expected valued decision, $x_t \in \mathcal{X}_{S_t}$.

Let $\phi_f(S_t)$ be the basis function evaluation f of state S_t at decision epoch t. We bin our features into the following three types: reward based, spatial based, and resource based. We normalize all basis function evaluations to scale all values to $[0, 1]$. This transformation facilitates direct basis function comparisons by examining value magnitudes.

All of our basis functions are interactions of more than one feature. For example, many of the functions below are an interaction of both a location and resource feature. We introduce the following feature types, then list the final basis functions at the end of the section. Each feature type is akin to a specific genre of information that ought to be considered during decision-making. We employ significant exploratory testing to develop the following basis functions and describe this exploratory process in Section IV.

The first feature type is reward-based. The intent for this feature is to help the system recognize the type of node on which the AUCAV is located when a reward is earned, and associate the earned reward with the node type it is currently on. Let

$$\phi^A_{NAI}(S_t) = \mathbb{I}_{\{\ell^A_t = \ell^{NAI}_{tn}\}} \tag{11a}$$

$$\phi^A_{HPT}(S_t) = \mathbb{I}_{\{\ell^A_t = \ell^{R,HPT}_{tr},\ \ \xi_r = 0\}} \tag{11b}$$

$$\phi^A_{HVT}(S_t) = \mathbb{I}_{\{\ell^A_t = \ell^{R,HPT}_{tr},\ \ \xi_r = 1\}} \tag{11c}$$

wherein the three features are aligned with the type of reward-bearing node on which the AUCAV is located. The three different reward bearing node types are: NAIs, HPTs, and HVTs. We construct the first three features as indicator functions ($\mathbb{I}$), which take on the value 1 when co-located with that reward-bearing node type and 0 otherwise. For example, if the AUCAV is co-located with a HVT, the indicator function $\mathbb{I}_{\{\ell_t^A=\ell_{tr}^{R,HPT},\ \xi_r=1\}}$ takes on the value 1.

The second feature type is spatial-based. Several key aspects of the spatial arrangement of the system are investigated; however, testing suggests the relative location of a node to the TAI and the location of a node relative to other nodes are of particular importance to aid decision-making.

We assume there is one TAI and it is rectangular in shape with four vertices. We employ the traditional cardinal direction indicators North (N), South (S), East (E), and West (W). Let the location of the centroid of the TAI be $\ell^{TAI,centroid} = (\left\|O^{SW} - O^{SE}\right\|, \left\|O^{NW} - O^{SW}\right\|)$. Let

$$\phi^A_{TAI}(S_t) = \frac{(dist^{max} - \left\|\ell^A_t - \ell^{TAI,centroid}\right\|)}{dist^{max}} \tag{12a}$$

denote the feature that returns a value as a function of the distance the AUCAV is from the TAI centroid, wherein the $dist^{max}$ is the hypotenuse of the entire battle space. The intent of this feature is to encourage the AUCAV to remain cognizant of its relative distance from the TAI as this is the area human intelligence officers have identified and labeled as a likely area of enemy discovery.

Another feature we consider is the relative distance each reward-bearing node is from the other reward-bearing nodes. We assume it is more lucrative to select a reward-bearing node that is relatively close to others and therefore attempt to include a notion of clustering into our features. We find that exhaustively enumerating the distances from each node to each other node is ineffective and inefficient. Additionally, partitioning the battle-space into sectors and introducing indicator functions is more efficient as it requires a smaller number of basis functions. However, its effectiveness is inadequate. To address relative distances into our ADP solution, we introduce an unsupervised machine learning algorithm first suggested by Ester *et al.* (1996) called Density-Based Algorithm for Discovering Clusters (DBSCAN) (Schubert *et al.*, 2017).

The DBSCAN algorithm is effective, efficient, and is aligned with the principle of model parsimony (Blumer *et al.*, 1987). The algorithm takes as its inputs two user-defined parameters and a list of points, then assigns a label for each point as one of three types. The node types are core, border, and noise. The two parameters are $\epsilon - Neighborhood$ and $minPoints$. A core point is a point which has at least the number of $minPoints$ within its $\epsilon - Neighborhood$. A border point is within a core point's $\epsilon - Neighborhood$ but it does not have $minPoints$ within its own $\epsilon - Neighborhood$. A noise point is not within the $\epsilon - Neighborhood$ of a core point. Introducing this algorithm into our basis functions requires two additional parameters to tune; however, we saw this as a worthy trade-off as we have the ability to condense density information into one of three labels. An abstract of the DBSCAN we employ is in Algorithm 1.

Algorithm 1 Abstract DBSCAN Algorithm

```
initialize value of $\epsilon - Neighborhood$
initialize value of $minPoints$
for Each point in set do count neighboring points within $\epsilon - Neighborhood$ if neighboring points $\geq minPoints$, label point as a core point
end for
Join Neighboring core points into clusters
for each non-core point do label as border point if within $\epsilon - Neighborhood$ of core point Otherwise, label as noise
end for
```

The output of this algorithm results in each considered point receiving a label of either core, border, or noise. Let p^{core}, p^{border}, p^{noise} denote a core, border, and noise point respectively. The following features indicate the node type on which the AUCAV resides.

$$\phi^A_{core}(S_t) = \mathbb{I}_{\{\ell^A_t = p^{core}\}} \tag{13a}$$

$$\phi^A_{border}(S_t) = \mathbb{I}_{\{\ell^A_t = p^{border}\}} \tag{13b}$$

$$\phi^A_{noise}(S_t) = \mathbb{I}_{\{\ell^A_t = p^{noise}\}} \tag{13c}$$

wherein indicator function $\mathbb{I}$ takes on the value 1 if a specific condition is met. For example, indicator function $\mathbb{I}_{\{\ell^A_t = p^{core}\}}$ takes on the value 1 if the AUCAV is co-located with a core point and is 0 otherwise.

The final feature type is resource-based. The resource of interest in the A3P is the amount of playtime ρ_t the AUCAV has at epoch t to execute the remainder of its mission. ρ^{max} is the maximum playtime the AUCAV begins a trajectory with. The amount of playtime the AUCAV expects to use for its next selected step is $\rho^{(\hat{\tau}(S_{t+1}|x_t)-\tau_t)}$. We then consider the following features:

$$\phi^{\,playtime}_{\,exp}(S_t) = \frac{\left(\rho_t - \rho^{(\hat{\tau}(S_{t+1}|x_t)-\tau_t)}\right)}{\rho^{max}} \tag{14a}$$

$$\phi^{\,playtime}_{\,prop\ remaining}(S_t) = \frac{\left(\rho_t - \rho^{(\hat{\tau}(S_{t+1}|x_t)-\tau_t)}\right)}{\rho_t} \tag{14b}$$

wherein $\phi^{\,playtime}_{\,exp}(S_t)$ provides the system the ability to be cognizant of the amount of playtime the next decision requires. The function subtracts the expected playtime expenditure from the current playtime remaining and standardizes over max playtime. This allows our model to learn that a decision that requires less playtime to complete is more appealing than a decision that may expend playtime frivolously. Moreover, $\phi^{\,playtime}_{\,prop\ remaining}(S_t)$ provides slightly different information as dividing by ρ_t allows the system to consider the *proportion* of playtime the next decision will expend as opposed to the amount. A larger proportion of remaining playtime remaining after the decision is more desirable as the irreplaceable playtime resource ought to be closely managed. Both of these basis functions together are required to capture the expected future value because of this subtle difference. For example, if the AUCAV were to operate on jet fuel, the first decision for a given trajectory may require a large amount of playtime but a small proportion to the remaining fuel left in the gas tank. Conversely, if the AUCAV is nearing the end of its mission with limited playtime remaining, every decision will likely account for a large proportion of its remaining playtime though they may be small amounts.

With the features defined, we introduce the basis functions used in our solution approach. Each basis function is an interaction of the features introduced above. Through extensive preliminary testing, we noticed the playtime resource was of critical importance to consider. Consequently, the resource features are the foundation to the basis functions with the spacial and reward based features as complimentary.

$$\phi_1(S_t) = \phi_{exp}^{playtime}(S_t)\ \phi_{HPT}^{A}(S_t) \tag{15a}$$

$$\phi_2(S_t) = \phi_{exp}^{playtime}(S_t)\ \phi_{HVT}^{A}(S_t) \tag{15b}$$

$$\phi_3(S_t) = \phi_{exp}^{playtime}(S_t)\ \phi_{TAI}^{A}(S_t) \tag{15c}$$

$$\phi_4(S_t) = \phi_{exp}^{playtime}(S_t)\ \phi_{border}^{A}(S_t) \tag{15d}$$

$$\phi_5(S_t) = \phi_{exp}^{playtime}(S_t)\ \phi_{core}^{A}(S_t) \tag{15e}$$

$$\phi_6(S_t) = \phi_1(S_t)\ \phi_{TAI}^{A}(S_t) \tag{15f}$$

$$\phi_7(S_t) = \phi_2(S_t)\ \phi_{TAI}^{A}(S_t) \tag{15g}$$

$$\phi_8(S_t) = \phi_1(S_t)\ \phi_{border}^{A}(S_t) \tag{15h}$$

$$\phi_9(S_t) = \phi_2(S_t)\ \phi_{border}^{A}(S_t) \tag{15i}$$

$$\phi_{10}(S_t) = \phi_1(S_t)\ \phi_{core}^{A}(S_t) \tag{15j}$$

$$\phi_{11}(S_t) = \phi_2(S_t)\ \phi^{A}_{core}(S_t) \tag{15k}$$

$$\phi_{12}(S_t) = \phi^{\,playtime}_{\,prop\ remaining}(S_t) \tag{15l}$$

$$\phi_{13}(S_t) = \phi^{\,playtime}_{\,exp}(S_t) \tag{15m}$$

Given the basis functions, we can now construct the value function approximation with the following linear approximation architecture

$$\bar{V}(S_t|\theta) = \sum_{f \in \mathcal{F}} \theta_f \phi^s_f(S_t) \equiv \theta^T \phi^s(S_t), \tag{16}$$

wherein $\theta = (\theta_f)_{f \in \mathcal{F}}$ is a column vector of basis function evaluation coefficients (weights) and $\phi^s(S_t)$ is a scaled column vector of basis function evaluations. Substituting the VFA in Equation 16 back into the Bellman Equation 9, we obtain

$$\bar{V}(S_t|\theta) = C(S_t) + \gamma^{(\hat{\tau}(S_{t+1})-\tau_t)} \mathbb{E}\big[\bar{V}(S_{t+1}|\theta)\big|S_t, X^{\pi}(S_t|\theta)\big], \tag{17}$$

wherein the decision x_t is determined via the decision function,

$$X^{\pi}(S_t|\theta) = \underset{x_t \in \mathcal{X}_{S_t}}{\operatorname{argmax}} \Big\{ C(S_t) + \gamma^{(\hat{\tau}(S_{t+1})-\tau_t)} \mathbb{E}\big[\bar{V}(S_{t+1}|\theta)|S_t, x_t\big] \Big\}. \tag{18}$$

With the VFA and decision function established, we discuss the manner in which the VFA, Equation 17, is updated and refined (i.e., statistical learning). The VFA is updated through an iterative process of sampling state-value pairs (or sample data). Our data are generated through simulation of our model given problem instance parameters. We employ a temporal difference approach to approximate and iteratively refine the value function (Bradtke & Barto, 1996). The following section introduces the algorithmic strategy, which controls the iterative statistical learning process of our VFA.

3.3.2 Algorithmic Strategy

To generate our ADP policy, we employ a finite sequence of steps shown in Algorithm 2. We employ an API approach because at the center of our algorithm is a loop derived from exact policy iteration. At the center resides the policy evaluation loop of size N. This inner loop evaluates a fixed policy (fixed θs) by collecting data through simulation in the form of state-value pairs. At the completion of a finite number of policy evaluation loops (inner loops), the m^{th} policy is produced using a least squares temporal differences (LSTD). A new policy θ^m is generated by merging the current policy $\theta^{\,m-1}$ with the sample policy $\hat{\theta}$. The smoothing of the two policies is managed by the learning rate parameter α.

Algorithm 2 API-LSTD Algorithm

```
Initialize $\theta$ (linear model coefficients or weights).
for $m = 1$ to $M$ do (Policy Improvement Loop) If $m > 2$, $r^{advPay} = 0$
    for $n = 1$ to $N$ do (Policy Evaluation Loop)
        Initialize problem instance to begin trajectory if $n = 1$ or $S_{t,n} = \Omega$.
        Generate a trajectory following next state $S_{t-1,n}$ (see Figure 9).
        Record basis function evaluation $\phi^s(S_{t-1,n})$.
        Employ $\epsilon$-greedy sampling as discussed in Section 3.3.3.
        Determine decision $x_t$ utilizing Equation 18.
        Simulate transition to next pre-decision $S_{t,n}$.
        Record contribution $C(S_{t,n})$ with $r^{advPay}$ as discussed in Section 3.3.4.
        Record discount factor $\gamma^{(\hat{\tau}(S_{t+1,n})-\tau_t)}$.
        Record basis function evaluation $\phi^s(S_{t,n})$.
    end for
    Update $\theta$ utilizing Equations 21-23.
end for
Return the decision function $X^{\pi^{LSTD}}(\cdot\ |\theta)$.
```

The algorithm begins with initializing values of θ at zero, which serves as the initial fixed policy to undergo evaluation. Next, we evaluate the performance of the current policy by randomly sampling states in a trajectory fashion and recording their accompanying values or basis function evaluations $\phi(S_{t-1,n})$. We then simulate one event forward in system time, determine the next action with Equation 18 (following an ϵ-greedy approach), and then record the basis function evaluations for the next state $\phi(S_{t,n})$. The collection of trajectory following state-value pair data continues for N iterations within the policy evaluation loop (inner loop). With a selected x_t, we simulate forward, and record the contribution $C(S_{t,n})$, current discount factor $\gamma^{(\hat{\tau}(S_{t+1,n})-\tau_t)}$ and basis function evaluations $\phi(S^M(S_{t,n}))$. The algorithm continues to the policy evaluation algorithm step with N temporal difference observations of sample data. We now have the information to update the approximate value of state S_t depicted in Equation 19 wherein $C(S_{t,n}) + \gamma^{(\hat{\tau}(S_{t+1,n})-\tau_t)}\theta\phi(S_{t,n}) - \theta\phi(S_{t-1,n})$ is the n^{th} temporal difference given the basis function weight vector θ.

$$\theta^\top\phi(S_{t-1}) = C(S_{t-1}) + \gamma^{(\hat{\tau}(S_t)-\tau_{t-1})}\mathbb{E}\big[\theta^\top\phi(S_t)|S_{t-1}\big] \tag{19}$$

A policy having finished N policy evaluation loops (inner loops) with its accompanied evaluation from N data, enters one m policy improvement (outer) loop. For each iteration of M outer policy improvement loops, we calculate a vector of estimates $\hat{\theta}$ of the existing but unknown coefficients θ through least squares regression. A $\hat{\theta}$ is sought that makes the sum of the temporal difference samples equal to zero. We denote the matrix form of the basis function evaluations, discounts, and contributions in Equation 20 wherein Φ_{t-1} and Φ_t are matrices consisting of basis function evaluations. Matrices Φ_t and Φ_{t-1} consist of N rows and $|\mathcal{F}|$ columns.

$$\begin{gathered}
\Phi_{t-1} \triangleq \begin{bmatrix} \phi(S_{t-1,1})^\intercal \\ \vdots \\ \phi(S_{t-1,N})^\intercal \end{bmatrix}, \quad \Phi_t \triangleq \begin{bmatrix} \phi(S_{t,1})^\intercal \\ \vdots \\ \phi(S_{t,N})^\intercal \end{bmatrix}, \\
\Gamma_t \triangleq \begin{bmatrix} \gamma^{(\hat{\tau}(S_{t+1,1})-\tau_{t,1})}\mathbf{1}_{\mathbf{1}\times|\mathcal{F}|} \\ \vdots \\ \gamma^{(\hat{\tau}(S_{t+1,N})-\tau_{t,N})}\mathbf{1}_{\mathbf{1}\times|\mathcal{F}|} \end{bmatrix}, \quad C_t \triangleq \begin{bmatrix} C(S_{t,1}) \\ \vdots \\ C(S_{t,N}) \end{bmatrix}
\end{gathered} \tag{20}$$

The rows of the discount factor matrix Γ_t are the discounts for the sample data, and the elements of vector C_t of length N are the recorded contribution values at each data observation n. We denote $\mathbf{1}_{\mathbf{1}\times|\mathcal{F}|}$ as a row vector of ones of length $|\mathcal{F}|$. We then use Equation 21 to calculate one sample estimate of θ.

$$\hat{\theta} = \left[(\Phi_{t-1} - \Gamma_t \odot \Phi_t)^\top (\Phi_{t-1} - \Gamma_t \odot \Phi_t) + \eta\mathbf{I}\right]^{-1} (\Phi_{t-1} - \Gamma_t \odot \Phi_t)^\top C_t \tag{21}$$

We denote $\eta\mathbf{I}$ as an $|\mathcal{F}| \times |\mathcal{F}|$ sized matrix wherein regularization parameter $\eta \geq 0$ serves as a means to avoid matrix inversion and over-fitting issues (Hastie *et al.*, 2001). With a new estimate of $\hat{\theta}$, we smooth the new estimate into the current estimate θ via Equation 23. Note that $\odot$ is the Hadamard product operator. We draw from Jenkins *et al.* (2021) and employ the learning rate (step-size rule) given in Equation 22 wherein $\beta \in (0, 1]$. This polynomial step-size rule greatly affects convergence of our algorithm and must be tuned. The magnitude of α_m decreases as m increases. The rate α_m decreases is controlled by algorithmic parameter β.

$$\alpha_m = \frac{1}{m^\beta} \tag{22}$$

Equation 23 depicts the smoothing process. The θ on the right-hand side of the arrow is the estimate based on previous iterations of the policy improvement loop $m-1$. The vector $\hat{\theta}$ is the estimate from the current iteration m.

$$\theta \leftarrow \alpha_m \hat{\theta} + (1-\alpha_m)\theta \tag{23}$$

The θ on the left-hand side is the new estimate and completes one iteration of the M policy improvement loops. The algorithm continues and produces a final estimate θ, after M policy improvement loops which parameterizes our policy (i.e., decision function) $X^{\pi^{LSTD}}(\cdot\ |\theta)$ for policy π^{LSTD}.

3.3.3 Sampling and Exploration

We choose to employ a trajectory following state sampling scheme and ϵ-greedy action sampling. We choose trajectory following (i.e., path following) for state sampling to approximately solve the A3P because the state space is uncountable. Many ADP solution approaches employ a sampling scheme wherein one random state sample is chosen per inner loop. The system is then simulated one decision epoch into the future to collect state-value pair data. Preliminary testing of the ADP solution approach to the A3P suggests that this single-step manner of sampling does not supply meaningful data for an effective solution. Indeed, we discourage single-step random sampling to solve the A3P approximately because we suspect such samples miss sampling the most relevant states recalling the uncountable state space. Consequently, we employ a trajectory following sampling scheme wherein, for each policy evaluation loop, we sample a sequence of steps always starting with a fixed deep zone entrance point with full playtime and ending with arrival to the deep zone exit point.

We implement an ϵ-greedy action sampling approach to balance exploration with exploitation of the ADP solution approach. Within every sample path generated in Algorithm 2, the ϵ-greedy approach encourages the system to choose the best decision it thinks available at step t 1-ϵ percentage of the time. This means that an AUCAV will choose to go to the next location that will earn maximum rewards 100(1-ϵ) percentage of the time and choose randomly from the set of feasible actions ϵ percentage of the time. The sampling of states through the random selection of actions may result in exploration of the state space that the system may not have conducted otherwise.

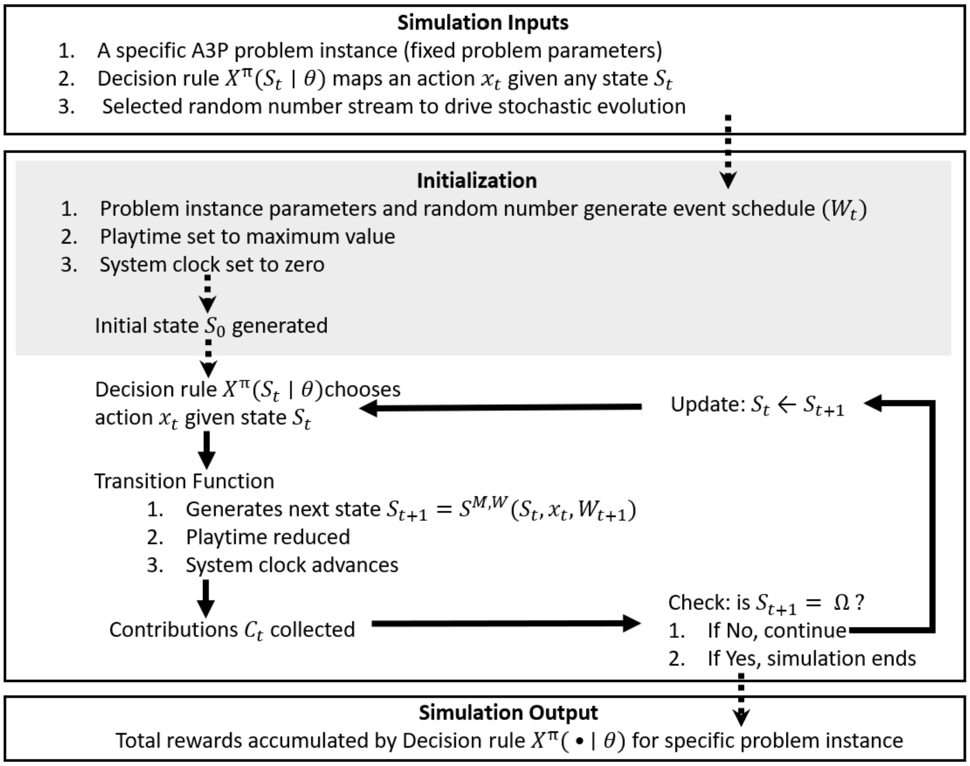
3.3.4 Reward Engineering

To solve the A3P approximately, we implement reward engineering. Also referred to as designing reward signals, we take great care in specifically designing the part of the AUCAV (i.e., entity or agent) environment that is responsible for computing the scalar reward received by the agent at every decision epoch (Sutton & Barto, 2018). We began our preliminary testing with having zero immediate reward received until the service is complete, meaning all epoch contributions (rewards) where a service is not complete, is equal to zero. This is justifiable as it aligns with the reality that an AUCAV's combat performance ought not to be rewarded for only planning to complete a service, but for actually completing a service. We realized that this could mean the ADP algorithm may not encourage learning in the manner we expected. For example, suppose during the progression of the learning algorithm a sample action was chosen either through ϵ-greedy sampling or that policy improvement loop's decision rule such that the next destination is an HVT. Indeed, Features 11a, 11b, and 11c exist to encourage the AUCAV to be cognizant that its choice results in a reward if it services that demand; however, if travel toward the HVT is interrupted by a new arrival, the AUCAV may never actually experience the reward as the service is not completed. We see this as a hindrance or obstacle to learning because this hypothetical AUCAV will never complete its service task due to the interruption, and therefore this decision, although appropriate, is not reinforced. As explained by Sutton & Barto (2018), reward signals ought to be designed so that as an agent learns, its behavior approaches, and ideally achieves, what the application's designer actually desires. In the case of the A3P, we assume the JFC wants targets destroyed and CCIRs answered. However, for the iterative ADP learning process to achieve such behavior, we construct a reward scheme that seeks to induce the desired behavior in an effective manner.

To overcome this suspected hindrance to learning, we employ a reward signal that gives r^{advPay} % of the reward of the chosen destination node upon selection. This reward is collected immediately within the epoch in which the reward-bearing node was chosen. For example, during the progression of Algorithm 2 a decision-rule within a given policy improvement loop selects an HVT as the next destination in epoch t, and the system receives an immediate reward of $r^{advPay}(r^{HVT})$ at epoch t. The system does not receive the full r^{HVT} unless the service is actually completed in a future decision epoch. This partial, advanced reward intends to encourage desired behavior of the AUCAV within the A3P system. We acknowledge there is risk given this algorithmic design as it consciously invites statistical bias into the algorithm. With such an engineered reward scheme, it is possible an AUCAV, within a specific A3P problem instance, may receive so many partial rewards that its incentive to actually complete services is insufficient to achieve the behavior that aligns with the JFC's intent. Indeed, in an extreme case, it is possible an AUCAV that learns under a reward scheme as we have introduced could fly an entire sortie without ever actually completing a service. This obviously does not align with our assumed JFC's intent. To monitor a given policy's alignment with behavior desired by our assumed JFC, we employ a simulation model to support the design and development of high-quality policies. In our simulation, this reward scheme does not exist; only serviced targets return a reward as outlined in Equation 7.

3.3.5 Simulation

In this section we introduce the simulation model used within the A3P solution approach. In an earlier section, we discussed the need to sample entire trajectories in lieu of one-step sampling to address the uncountable and continuous state space of the A3P. The following simulation model serves as the engine behind the trajectory-following scheme fueling our learning algorithm. Sampling resulting from simulation-driven trajectories enables the collection of effective samples of state-value pairs. The simulation model also serves as a mechanism for evaluating any given policy to include the benchmark, DROP policy. See the graphical flow of the simulation model in Figure 9.



Simulation Inputs
1. A specific A3P problem instance (fixed problem parameters)
2. Decision rule $X^{\pi}(S_t \mid \theta)$ maps an action x_t given any state S_t
3. Selected random number stream to drive stochastic evolution
Initialization
1. Problem instance parameters and random number generate event schedule (W_t)
2. Playtime set to maximum value
3. System clock set to zero
Initial state S_0 generated
Decision rule $X^{\pi}(S_t \mid \theta)$chooses action x_t given state S_t
Update: $S_t \leftarrow S_{t+1}$
Transition Function
1. Generates next state $S_{t+1} = S^{M,W}(S_t, x_t, W_{t+1})$
2. Playtime reduced
3. System clock advances
Contributions C_t collected
Check: is $S_{t+1} = \Omega$?
1. If No, continue
2. If Yes, simulation ends
Simulation Output
Total rewards accumulated by Decision rule $X^{\pi}(\bullet \mid \theta)$ for specific problem instance

Figure 9. Graphical Depiction of Simulation Model

Each simulation requires as input a specific A3P problem instance, a decision-rule, and a random number stream. A single problem instance implies problem parameters are fixed. For example, arrival rate of new enemy targets λ and initial amount of AUCAV playtime ρ^{max}, is specified. The decision rule is the current policy to undergo sampling, and the random number seeds allow the simulation user to control multiple samples from a single problem instance as well as the ability to use common random number generation to reduce variance when comparing performance of policies. The discrete event simulation is driven by an event schedule constructed a priori upon initialization. The event schedule is a list used to determine what event happens next in a discrete event simulation (Banks *et al.*, 2013). In the A3P the list of new target arrivals is generated at onset of the simulation with pre-determined arrival time, location, and type. As the system evolves, the transition function uses this event list and compares the time for each event with the travel time for the AUCAV from its current position to its intended destination. The process that takes the least amount of time, either new arrival or the AUCAV reaching its decided destination, determines the next decision epoch. The simulation then continuously evolves the system, collecting rewards and reducing playtime until the only feasible decision to select next is the deep zone exit point ($x_t = \Omega$). Once the system evolves to this final state, the simulation terminates. In Algorithm 2, the system may re-initialize to collect the pre-determined number N of algorithmic samples (state-value pair data). For all evaluation simulations in this research we utilize a discount rate of $\gamma = 1$. With $\gamma = 1$, we refer to ETDR as mean total reward. When we use this simulation to determine the quality of a given decision-rule, one trajectory is sufficient per random number stream.

IV. Testing, Analysis, and Results

In this chapter we introduce a representative scenario of the autonomous attack aviation problem (A3P) that serves as the foundation for our quantitative analysis of policy performance. We demonstrate the efficacy of our ADP policy using specific, realistic problem instances of the A3P that serve as a means to quantitatively measure and compare performances of each policy outlined in Chapter III. We design and conduct computational experiments to tune the algorithm hyper-parameter values of the ADP solution procedure to maximize performance. Moreover, we conduct case study analysis using common random number simulations to gain insights on policy performance. The computational experiments are conducted utilizing an Intel Core i7, 2.20GHz, 4-core processor with 16GB of RAM and MATLAB (2020b) parallel processing toolbox. To determine solutions for the benchmark DROP policy, we utilize IBM's CPLEX version 12.9.0.

4.1 Representative Scenario

We develop a notional A3P scenario wherein a single autonomous unmanned combat aerial vehicle (AUCAV) is tasked to conduct a strike coordination and reconnaissance (SCAR) mission in a deep zone given a fixed criticality, accessibility, reputability, vulnerability, effect, and recognizability (CARVER) tool, an initial threat template (recall Figure 7), and named area of interest (NAI) reconnoiter requirements. The area of operations (AO) for this mission exists on terrain of scale and makeup similar to that found at the National Training Center (NTC) at Fort Irwin, California (USGS, 2021). The terrain is arid with near zero vegetation and permissive to armored ground vehicle maneuvers save intermittent, steep, elevation changes serving as obstacles to ground movement and protection, see Figure 10 from Coryell

& Heap (2016). This terrain type is selected to provide smooth transfer of analysis and insights presented herein to potential application in terrain familiar to modern war-fighters.

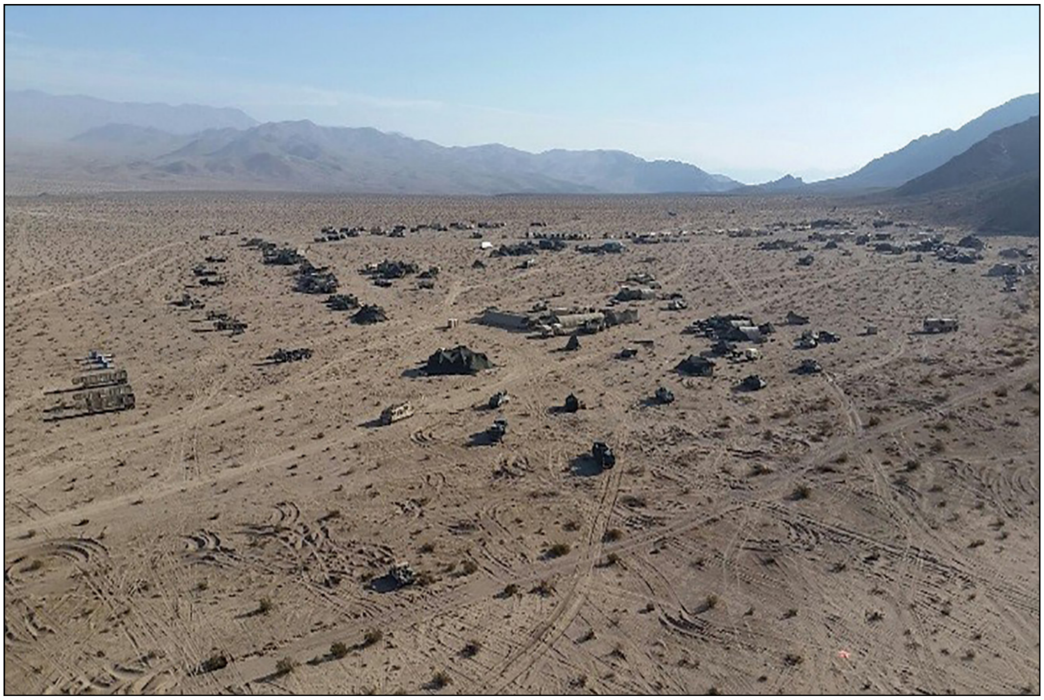


Figure 10. Aerial View of Military Vehicles at the NTC (Coryell & Heap, 2016)

The notional US Army Division current operations section is the controlling authority of the AUCAV. Its duty is to align AUCAV behavior to the Joint Forces Commander's (JFC's) intent as received, in part, in the form of scalar value inputs to the contribution function (see Equation 7). The JFC is responsible for battlefield success and approves the employment of the AUCAV enabled sortie through certifying the reward values for destruction of high value targets, high payoff targets, and answering commander's critical information requirements (CCIRs) through visitation of NAIs. Our notional JFC approves the following guidance in Table 3 to facilitate employment of the AUCAV.

The division targeting working group drives the decide, detect, deliver, assess (D3A) targeting cycle. The cycle is dynamic and stochastic with a new target arriv-

Table 3. JFC's Intent

$$\begin{aligned} r^{HVT} &= 1000 \\ r^{HPT} &= 10 \\ r^{CCIR} &= 1 \end{aligned}$$

ing to the high payoff target list (HPTL) at rate λ (example HPTL in Figure 5). For example, a $\lambda = \frac{1}{10}$, means on average, a new enemy target arrives to the HPTL once every 10 minutes. The AUCAV speed and playtime are aligned to estimates found in sources covering current US Army acquisition pursuits such as Judson (2020b), Judson (2019b), and Freedberg (2021). This mission occurs in the context wherein a US Army Division conducts a high operational tempo, direct action, large-scale combat operation (LSCO) with one, near-peer adversary whose composition, disposition, and behavior is aligned with an armored, Brigade Tactical Group (BTG) as found in Department of the Army (2011b) and Department of the Army (2011a). This representative scenario is similar to two historical case studies utilizing attack aviation. The first is the US Army deep attack on the *Medina* Division's armor and artillery in the Spring 2003 (Kem, 2018), and the other is the first US Army Apache deep attacks of Desert Storm in 1991 intended to destroy adversary anti-aircraft assets (Smithsonian, 2015).

The AO is 100 kilometers wide by 100 kilometers deep with the friendly forces arrayed to the south. We assume the forward line of troops (FLOT) of friendly ground forces is south of the A3P AO. The AO is indexed with a military grid reference system (MGRS) with coordinate (0,0) denoting the very center of the square AO. The deep zone exit point is aligned with the southern boundary of the AO and 5 km to the west of center (-5,-50). The AUCAV starting position is also aligned with the southern boundary and aligned 5 km to the east of center (5,-50). Figure 11 depicts an A3P SCAR Mission AO constructed in the MATLAB computing language.

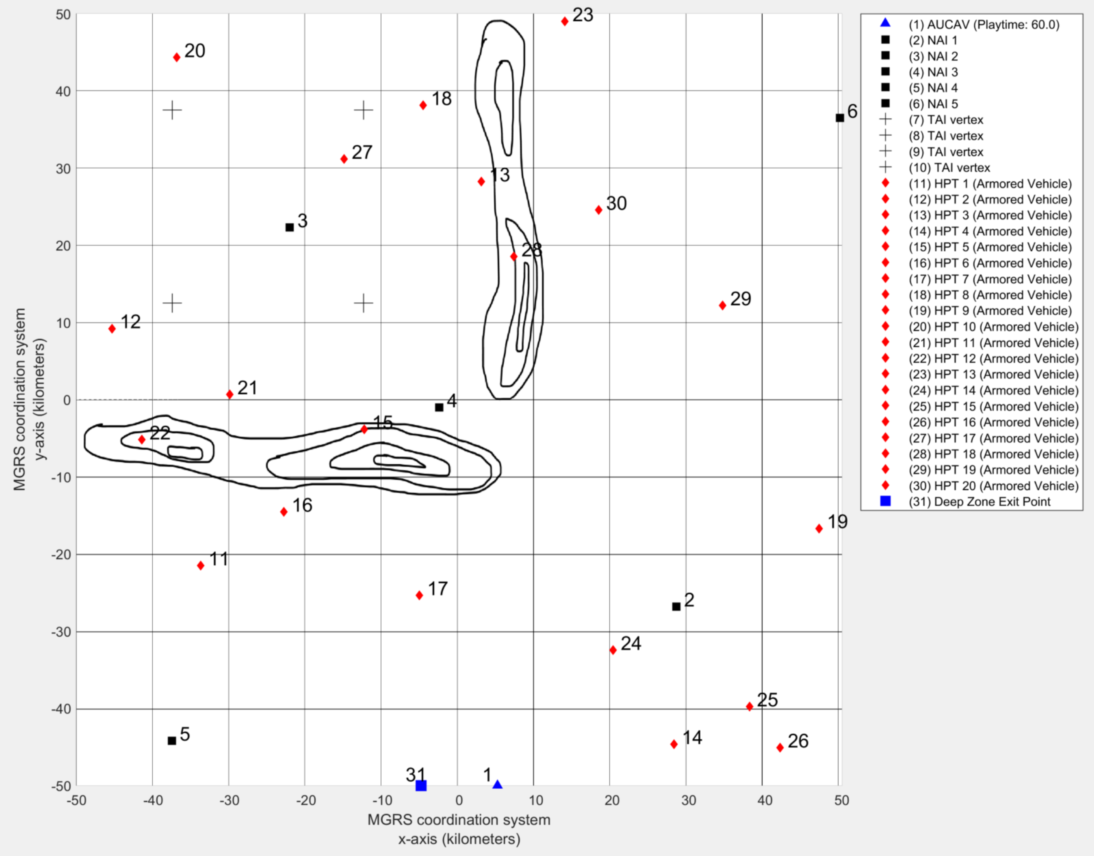


Figure 11. SCAR Area of Operations in MATLAB

In this baseline A3P instance, blue forces have constructed a list of 20 known high payoff targets (HPTs) arrayed uniformly across the 100 km by 100 km space. In addition, the JFC values the visitation of five NAIs. This is the known, initial HPTL and NAI requirements that invoke the AUCAV SCAR mission launch. Through information preparation of the battlefield (IPB), the intelligence officers of the division expect more targets to manifest to the northwest, behind the L-shaped ridge line. Intelligence officers also expect a 20% chance that any possible future arrivals will be high value targets (HVTs). Such HVTs may be specialty assets such as radar, air defense, artillery, or breaching assets. As a result, the division staff establishes a square shaped target area of interest (TAI) with four vertices at (-37.5,37.5), (-37.5,12.5), (-12.5, 37.5), and (-12.5,12.5). This square-shaped TAI centroid is located at (-25,25).

The enemy forces consist of a BTG conducting a large-scale, offensive operation from north to south. As expected by blue forces intelligence estimates, enemy ground commanders array a majority of their forces in the northwest of the SCAR mission AO behind the L-shaped ridge line. A majority of red forces locate to the northwest to consolidate, organize, and conceal their ground forces in a natural assault position for a future, synchronized brigade attack. The starting state of this baseline instance of the A3P is displayed in Figure 11.

4.1.1 Problem Instance Factor Selection

In this section we introduce and define the problem factors and our reasons for selecting the items of information that are experimental factors. Moreover, we choose factors whose values we vary to generate specific problem instances for deeper study. Table 4 shows the list of A3P problem factors, categorized into factors we choose to study in depth and factors we choose to fix. Factors we study are those factors whose values we modify to investigate policy robustness.

An A3P with a particular set of problem factor values is a problem instance. We study robustness of policies by modifying the value of a problem factor of interest, generating a new problem instance and observing the impact on quality of a given policy. A policy that provides high-quality solutions to multiple problem instances earns the title of robust. Study factors are selected intentionally since their inclusion in our experimentation comes at a cost of computation time, particularly for the benchmark policy. Each additional factor and factor level can greatly increase computation time, so only those problem factors whose study is expected to provide the most insight are selected. Fixed value problem factors are listed in the bottom of Table 4. We record their values and our intentions behind their fixed values to increase understanding and facilitate reproduction.

Table 4. Problem Factors

Problem Factors to Study	Factor Levels
Target Arrival Rate per 10 mins, λ	1, 3
AUCAV Playtime, ρ	60 min , 120 min
Shape of battlespace	Linear, Non-Linear (See Figure 2)
Fixed Factors	Factor Level
AO Characteristics	100 km by 100 km
AUCAV Speed	108 Knots
Number of NAIs	5
Number of Targets on Initial HPTL	20
Percentage of new target arrivals as HVTs	20%
Geographically asymmetric HPT arrival	95% northwest
JFC's intent (reward values)	See Table 3

We choose to vary the rate of arrival as the inherent uncertainty and variability of enemy behavior is of primary concern in war-fighting. The two levels of enemy arrivals intend to simulate A3Ps of varying intensity with regard to enemy activity. We choose to vary playtime because time is a resource that is normally of critical importance and must be closely managed. Although the overall flying playtime capacity of a AUCAV may exceed 120 minutes, we seek to investigate varying mission lengths. Indeed, mission duration is not only constrained by fuel capacity, other factors impact it as well, such as a changing battlefield environment or changes in JFC priorities, which occur frequently in contemporary armed conflict. Finally we choose to vary the shape of the battlespace to simulate the various terrains in which an A3P may exist. Recall the linear and non-linear AOs in Figure 2.

4.2 The Benchmark Policy

The purpose of a benchmark policy is to provide a standard from which to measure the performance of our ADP solution approach. We do not have the option of measuring quality relative to an optimal policy since the A3P has an uncountable state-space and therefore precludes current methods to calculate an optimal policy. Instead, we measure the solution quality of our ADP policy through comparison with a benchmark policy. We propose the deterministic, repeated, orienteering problem (DROP) as the benchmark solution method for this research. The DROP provides a suitable benchmark policy because, when given a set of nodes to visit with known rewards and playtime remaining, the DROP will return the optimal route solution within any given decision epoch t. We suggest that the area of artificial intelligence and autonomy is still in its formative years and there are no widely accepted rules or policies from which to compare autonomous behavior within the A3P. Given this, the DROP policy seems reasonable to embed in an autonomous agent found within the A3P. In addition, the orienteering problem is well established in literature and can expand to multi-AUCAV instances as discussed in Chapter II with the team orienteering problem (TOP).

The mathematical formulation of the DROP consists of classical components found in optimization literature: objective function, decision variables, and constraints. The DROP policy is determined by repeatedly solving a mixed integer linear program (MILP) based on the TOP formulation found in Gunawan *et al.* (2016). The π^{DROP} policy takes as its input state S_t, calculates a routing solution, and identifies the first stop in the route as a decision x_t. The system then evolves to S_{t+1}, and the π^{DROP} solves again until the AUCAV's arrival to the exit point, Ω.

The MILP seeks to maximize the collected reward for a given state S_t constrained by the playtime remaining ρ_t through planning the optimal route. A route starts with the current AUCAV location, travels through reward-bearing nodes to maximize collected reward, and ends at the deep zone exit point. Let X_{ij} denote the decision variable taking the value 1 if the decision is to travel from node i to node j, and 0 otherwise. The following formulation depicts the mathematical model of the DROP solution approach at one epoch t

$$\textbf{Obj}: \max_{\boldsymbol{X_{ij}}} \sum_{i=2}^{(|N|-1)} \sum_{j=2}^{|N|} P_i X_{ij}, \tag{24}$$

$$\text{s.t.} \sum_{j=2}^{|N|} X_{1j} = \sum_{i=1}^{(|N|-1)} X_{i|N|} = 1, \tag{25}$$

$$\sum_{i=1}^{(|N|-1)} X_{ik} = \sum_{j=2}^{|N|} X_{kj} \leq 1; \text{ for } k = 2, ..., (|N|-1) \tag{26}$$

$$\sum_{i=1}^{(|N|-1)} \sum_{j=2}^{|N|} \rho_{ij} X_{ij} \leq \rho_t, \tag{27}$$

$$2 \leq u_i \leq |N|; \text{ for } i = 2, ..., |N|, \tag{28}$$

$$u_i - u_j + 1 \leq (|N|-1)(1 - X_{ij}); \text{ for } i = 2, ..., |N| \tag{29}$$

$$X_{ij} \in \{0, 1\}, \forall \; i, j \in N \tag{30}$$

wherein $N = \{1, ..., |N|\}$ denotes the set of nodes including the current AUCAV location, the A3P exit location, and all NAIs and targets requesting service in state S_t at epoch t. Let P_i denote the payoff or reward for visiting node i. Define the AUCAV location $\ell_t^A = 1$ and the deep zone exit location $\Omega = |N|$. All other nodes are reward-bearing nodes within the battlespace, including NAIs, HPTs, and HVTs. Let $\{i + 1, ..., (|N| - 1)\} \equiv \{(\ell_{tn}^{NAI})_{n \in \mathcal{N}_t} \bigcup (\ell_{tr}^{R,HPT})_{r \in \mathcal{R}_t}\}$. Let u_i, in the subtour constraints (i.e., Constraints 28 and 29), denote the position of the visited node i in the route.

The objective seeks to maximize the total reward. Constraint 25 ensures the route starts at the current AUCAV location and ends at the deep zone exit point.

Constraint 26 ensures each node is visited only once for a given route and the nodes of the route are sequential. Constraint 27 constrains the route length given the current playtime available at epoch t, wherein ρ_{ij} indicates the playtime used to travel from node i to node j and must be less than the playtime remaining ρ_t. Constraints 28 and 29 are subtour prevention constraints. Constraint 30 enforces non-negativity.

Perhaps the most powerful aspect of the DROP is that for every state S_t, the policy considers all feasible routes and chooses not only the next best node destination, but every subsequent node thereafter to maximize rewards in the form of an optimal route. This result, however, comes at a computational cost, which we measure and report. Indeed, the DROP requires much more time to arrive at its solution than the approximate solutions. Moreover, the DROP policy does not anticipate new target arrivals, but its effectiveness is still very respectable given its ability to plan an entire route. This is the primary distinction we seek to investigate – to compare the performance between the full route-planning DROP policy to our ADP policy that anticipates the possibility of new enemy target arrivals.

4.3 Experimental Design

The purpose of our experimentation campaign is to measure the performance of our API-LSTD algorithmic solution approach relative to a benchmark and attempt to discern why the ADP policy behaves the way that it does. We define performance as a function of the solution quality, computational effort, and robustness (Barr *et al.*, 1995). We measure solution quality in the mean total reward earned by a given solution over 30 simulation runs of a given problem instance. A single run of a simulation consists of the total rewards collected by policy π with a single use of playtime, ρ^{max}. Solution quality of the π^{LSTD} is measured relative to the π^{DROP} benchmark. We measure computational effort as the number of minutes required to find a solution to 30 simulation runs of one problem instance of the A3P. Moreover, we determine robustness through comparing performance on varying problem instances. A policy earns the title robust when it performs well across multiple problem instances.

Our experimental campaign consists of three phases: preliminary testing, designed experimentation, and targeted case study investigation. The primary focus of preliminary testing is on π^{LSTD} basis function construction whereas the next phases of the experimentation campaign focus on identifying and tuning the algorithmic parameter values conducive to returning quality solutions. The case study investigations involve revisiting scenarios in which the DROP and ADP policies perform differently to discern the root cause of the disparity in behavior through recreating the specific, tactical conditions. This final phase of experimentation fuels our insights and conclusions.

4.3.1 Preliminary Testing and Insights

We begin our experimentation campaign with preliminary testing to explore the operating space of the A3P and scope a meaningful experimentation region of interest. Besides the number and makeup of basis functions, we explore the seven algorithmic factors in Table 5.

Table 5. Algorithmic Factors Under Investigation

N	number of policy evaluation loops
β	learning rate parameter
η	regularization parameter
ϵ-greedy	action space search
r^{advPay}	reward engineering signal
ϵ-Neighborhood	DBSCAN parameter
minPoints	DBSCAN parameter

We follow a structured process during preliminary testing to organize our investigation. We depict our preliminary testing process in Figure 12.

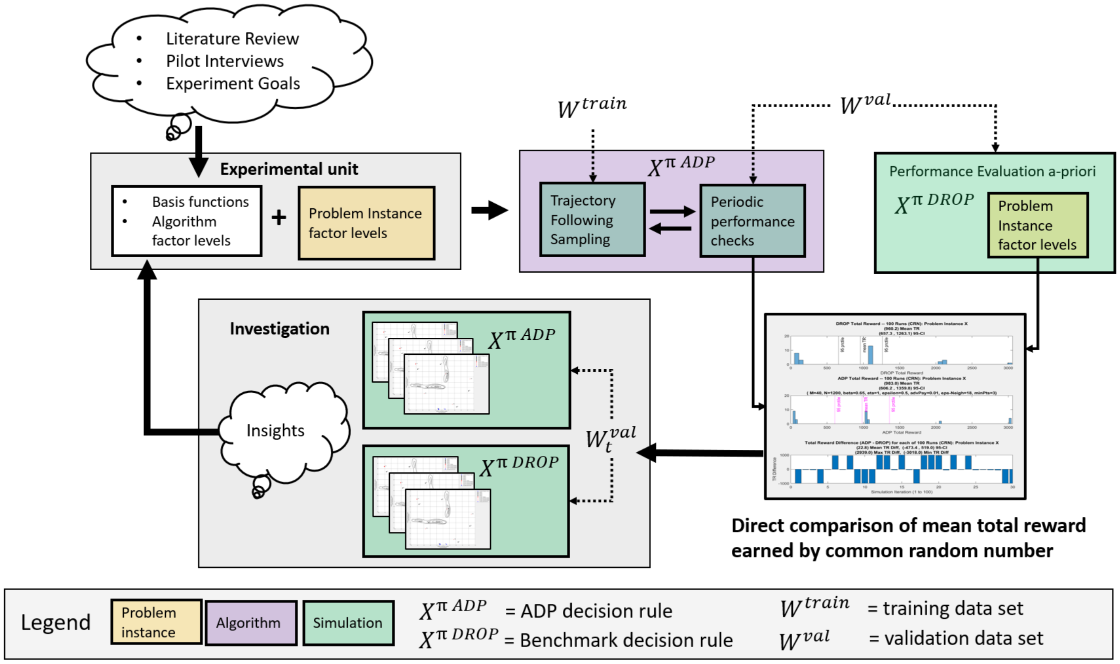

Literature Review
Pilot Interviews
Experiment Goals
W^{train}
W^{val}
Experimental unit
Basis functions
Algorithm
factor levels
+
Problem Instance factor levels
$X^{\pi\ ADP}$
Trajectory Following Sampling
Periodic performance checks
Performance Evaluation a-priori
$X^{\pi\ DROP}$
Problem Instance factor levels
Investigation
$X^{\pi\ ADP}$
Insights
W_t^{val}
$X^{\pi\ DROP}$
Direct comparison of mean total reward earned by common random number
Legend
Problem instance
Algorithm
Simulation
$X^{\pi\ ADP}$ = ADP decision rule
$X^{\pi\ DROP}$ = Benchmark decision rule
W^{train} = training data set
W^{val} = validation data set

Figure 12. Preliminary Testing and Investigation Process

The use of common random numbers (CRN) is synchronized, meaning each random number used in one model for some purpose is used for the same purpose in the second model (Banks *et al.*, 2013). This means both policies, the benchmark and the ADP, for a fixed problem instance, face the same stochastic enemy arrivals, same locations, and same times for their performance evaluation simulations (recall Figure 9). This allows us to assume that the difference in the total reward earned by a policy for a given problem instance, and given CRN, is due to the effectiveness of the decision rule in making sequential decisions under uncertainty. Any disparity in policy performance for a given problem instance and given CRN is not blamed on the inherent randomness in the problem environment.

Preliminary testing is extensive. A critical output of the preliminary testing process is the attainment of an ADP policy whose basis functions and algorithm parameter values achieve a higher mean total reward than the DROP for a given problem instance. This solution quality suggests that the basis functions are satisfactory, and we can reasonably progress to a more thorough investigation and tuning of algorithmic factor values through designed experimentation. We use this initial case study as a foundation for designed experimentation of algorithm parameter value searches. The case we choose to serve as the foundation of our screening and follow up experiments is shown in Figure 13.

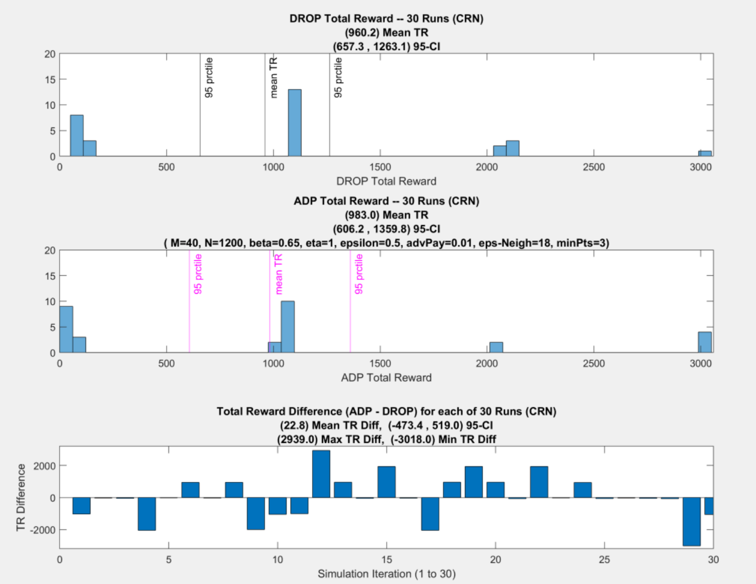


Figure 13. Baseline ADP Performance for Testing

All data shown in Figure 13 is policy performance on A3P Problem Instance 3. The top third of Figure 13 shows a histogram of the scores achieved by the π^{DROP} benchmark policy over 30 runs with accompanied mean value and 90% confidence interval. The middle third of Figure 13 shows π^{LSTD} performance. Finally, the bottom third of Figure 13 shows a stacked bar chart of the differences of policy scores arrayed by the 30 runs. This figure is useful for investigation as it depicts which specific runs have a disparity in policy performance and display the bars proportional to the magnitude of performance difference. If the dark blue bar is above the horizontal line, the ADP achieved a superior mean total reward for that run.

4.3.2 Screening Designed Experiment

We design our screening experiment based on the ADP policy and problem instance that return a mean total reward higher than that of the benchmark from Problem Instance 3 in Figure 13. This initial ADP performance establishes a foundation on which to base a screening experiment to capture the best combination of LSTD parameter settings that provide the best π^{LSTD} performance possible. We investigate robustness by designing eight problem instances and apply algorithmic factor levels similar to the ones found in Figure 13. We include first order interactions of the basis functions, second order, and third order terms with a total of $|\mathcal{F}| = 134$. The intent of this 2^6 full factorial designed experiment is to identify the superlative algorithmic factor levels for each of the eight A3P problem instances that achieve the greatest mean total reward compared to all other combinations of algorithmic factor levels. The factor levels for this designed experiment are shown in Table 6. We keep parameter η fixed at 1.

Table 6. Designed Experiment Factors and Levels

Category	Factor	Parameter Settings
Problem Instance	Battlespace Shape	{Linear AO, Non-Linear AO}
	Arrival Rate λ	$\{\frac{1}{10min}, \frac{3}{10min}\}$
	Playtime ρ^{max}	{60, 120}
API-LSTD	N	{1000, 5000}
	β	{0.5, 0.6}
	ϵ-greedy	{0.3, 0.5}
	r^{advPay}	{0, 0.01}
	ϵ-Neighborhood	{12.4, 18}
	$minPoints$	{3, 4}

The eight problem instances with their accompanying problem factor levels are shown in Table 7. The problem instance parameter levels are selected to provide sufficient insight to investigate a region of experimentation within the operational region of the A3P. We seek to gain a foothold of research insights most useful for this debut research of the A3P.

Table 7. Eight Problem Instances

	Problem Instance Parameter Settings		
Instance	Battlespace Shape	10λ	ρ^{max}
1	Linear	1	60
2	Linear	1	120
3	Linear	3	60
4	Linear	3	120
5	Non-Linear	1	60
6	Non-Linear	1	120
7	Non-Linear	3	60
8	Non-Linear	3	120

4.4 Experimental Results

The designed experiment returns the superlative π^{LSTD} parameters settings for each problem instance shown in Table 8. For six of the eight problem instances, the superlative parameter setting included the r^{advPay}, suggesting that this parameter is useful for solving the A3P approximately.

Table 8. Superlative ADP Parameter Values

	Superlative API-LSTD parameter settings by problem instance							
Instance	M	N	β	η	$\epsilon - greedy$	r^{advPay}	$\epsilon - Neigh$	$minPts$
1	30	5000	0.6	1	0.5	0	18	4
2	30	1000	0.6	1	0.5	0.01	18	3
3	20	1000	0.5	1	0.3	0.01	18	4
4	10	5000	0.5	1	0.5	0	12.4	4
5	10	1000	0.5	1	0.5	0.01	12.4	3
6	10	5000	0.5	1	0.3	0.01	18	4
7	50	1000	0.5	1	0.3	0.01	18	3
8	50	1000	0.5	1	0.3	0.01	12.4	4

Table 9 depicts the solution qualities of the π^{LSTD} and π^{DROP} policies. The far right column of the table shows the mean differences and 90% confidence interval half-widths for the performances and provides insight into quality and robustness. In terms of solution quality, the data suggest the π^{DROP} performs more robustly in terms of quality than the π^{LSTD} of the eight problem instances investigated. However, for Problem Instances 1 and 5, the π^{LSTD} does outperform the π^{DROP} with respect to quality. Indeed, the π^{LSTD} superior performance is impacted by two problem factor levels, $\frac{\lambda}{10} = 1$ and $\rho^{max} = 60$. This result suggests that the JFC may prefer the π^{LSTD} performance quality over the π^{DROP} when a mission is of shorter duration and new enemy arrivals during a mission are fewer.

Table 9. Performance (Quality and Robustness)

	π^{DROP}			π^{LSTD}			Mean Difference (ADP - DROP)		
Instance	Mean TR			Mean TR			Mean TR		
1	229.27	±	103.9	471.4	±	173.1	242	±	210.9
2	1652.3	±	309.7	1405	±	381.3	-246.7	±	514.1
3	960.2	±	251.6	916.2	±	300.5	-44	±	404.5
4	5170	±	793.1	2411.1	±	448.7	-2758.8	±	804.3
5	571.3	±	192.9	843.6	±	238	272.3	±	303.1
6	1884.1	±	365.3	1752.5	±	364.9	-131.6	±	558.9
7	1989.9	±	410	1781.8	±	427.6	-208.1	±	665.8
8	5950.8	±	755.6	4400.3	±	499.6	-1550.5	±	965.8

Table 10 depicts policy computational effort or speed of attaining solutions. The data suggest the π^{LSTD} is far superior to the π^{DROP} in terms of performance speed. The π^{LSTD} is between 10 to 216 minutes faster at calculating a solution to 30 runs of a problem instance as compared to the π^{DROP}. The performance speed gap increases substantially with increased $\frac{\lambda}{10}$ and ρ^{max}. Indeed, this is the reason only 30 runs of each problem instance are investigated. When comparing performance values between the π^{LSTD} and π^{DROP}, the benchmark values are calculated only once, with subsequent investigations read from a table. We encourage future researchers to budget sufficient time to investigate performance over 100 runs for increase quality data accuracy.

Table 10. Performance of 30 Simulation Runs (Speed and Robustness)

	Benchmark Computational Effort	LSTD Computational Effort	Computational Effort difference (ADP - DROP)
Instance	(mins)	(mins)	(mins)
1	25.5	0.33	-25.17
2	19.9	0.78	-19.12
3	196.93	0.59	-196.34
4	185.5	2.06	-183.44
5	11.25	0.29	-10.96
6	23.04	0.55	-22.49
7	91	0.76	-90.24
8	218.07	2.07	-216

The results of the 2^6 full factorial experiment suggest that overall performance of the π^{LSTD} is contested with the π^{DROP} given the problem instance factor levels and LSTD algorithmic factor levels under investigation. A decision-maker valuing solution quality may prefer π^{DROP} with respect to quality; however, the π^{LSTD} performance quality for problem instances with low $\frac{\lambda}{10}$ and ρ^{max} exceeds π^{DROP} quality. In addition, these results suggest that the π^{LSTD} far surpasses the π^{DROP} performance with respect to computational effort. We continue our investigation into the third phase of our campaign of experimentation, performing a specific case study analysis to discern why the policies seem to behave the way they do.

4.5 Case Study Investigation

Our case study investigation begins by selecting certain behaviors we expect are most meaningful for decision-makers and future research. The primary question we choose to investigate is: Why did the π^{LSTD} outperform the π^{DROP}? Once we gain insight into the cause, we seek to confirm our newly formed hypothesis with additional investigations into other problem instances. To address these inquisitions, we focus our attention on policy performance quality in Problem Instance 1.

4.5.1 Problem Instance 1

The results suggest a relationship with π^{LSTD} increased attainment of mean total reward over the π^{DROP} in Problem Instances 1 and 4. We seek to discern the possible relationship between this performance and the problem factor levels of enemy target arrival and playtime. Problem Instance 4 seems to support suspicion that this increase in policy quality is robust considering the battlespace shape problem factor. We begin our investigation by considering the performances of the 30 simulations of Problem Instance 1 shown in Figure 14. This figure alone suggests the π^{DROP} never successfully destroyed more than 1 HVT in any of the 30 runs as no scores are over 1200.

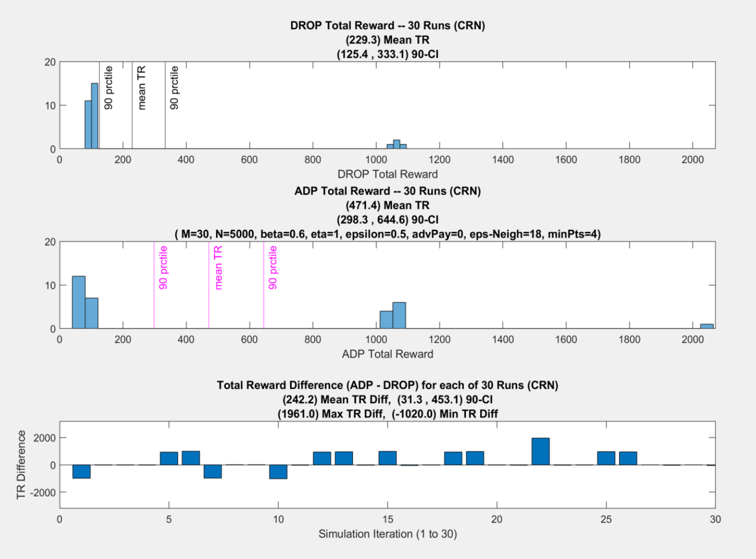


Figure 14. Problem Instance 1 Performance Quality

The data in Figure 14 suggest π^{LSTD} is superior in Runs 5,6,12,13,15,22,25, and 26. We choose to investigate performance in Run 22 (CRN seed 122) since the magnitude of the performance gap seems to be the greatest. Figure 15 depicts the initial starting state both the π^{DROP} and π^{LSTD} experience. They experience the same arrivals at the same times, but the sequential decision-making they perform changes the outcomes.

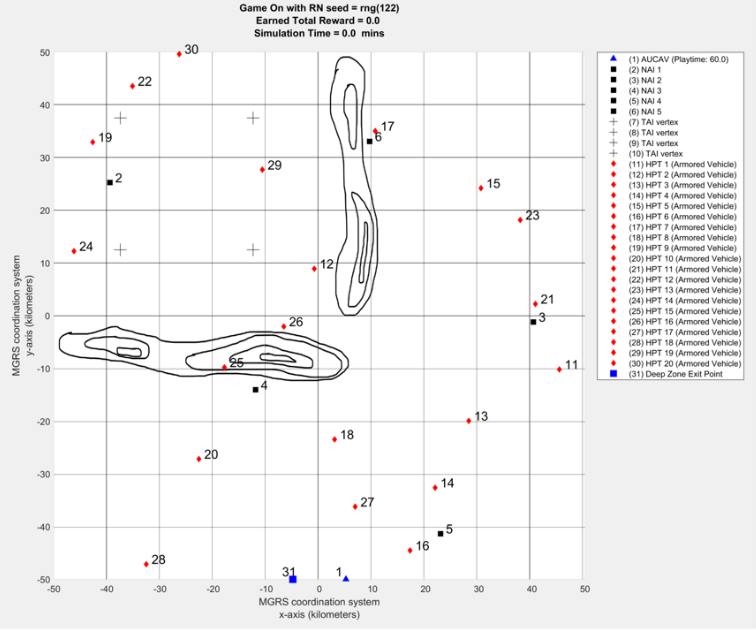
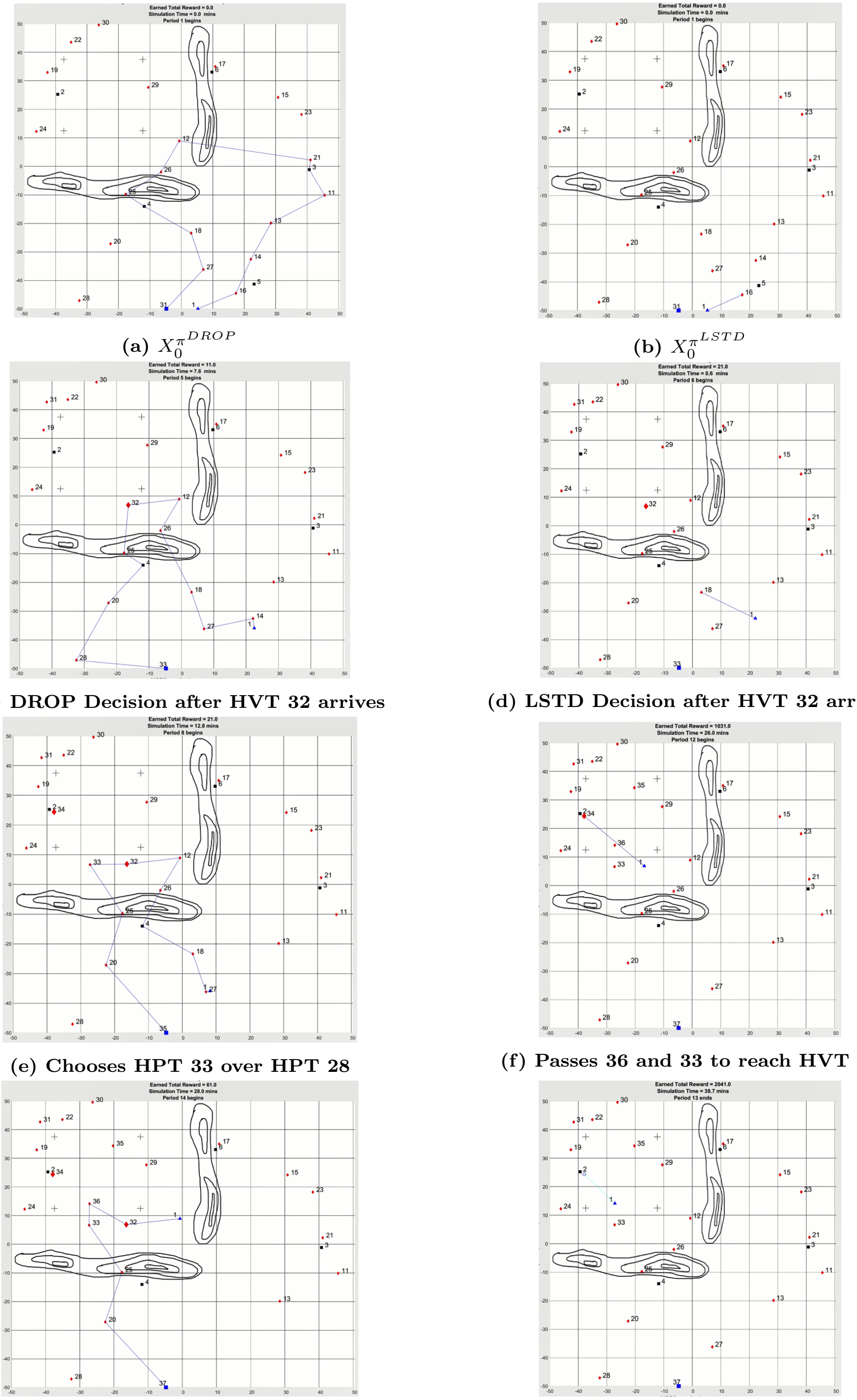


Figure 15. Problem Instance 1 at State S_0

There is a high density of initial targets and NAIs to the southeast of the AO. This is random since the initial set of HPTs and NAIs is uniformly distributed. Only the new arrivals are subject to asymmetry favoring the northwest. We omit the legend on the right-hand side of the maps in Figure 16 for efficiency.

Figure 16 depicts a similar process to the preliminary testing outlined in Figure 12. We re-load Run 22 of Problem Instance 1 (CRN seed 122) to step through the flight path of each policy to gain insight. The sequence of figures depict specific systems states while implementing the π^{DROP} and π^{LSTD} policies in the left and right columns, respectively. In Problem Instance 1 (CRN seed 122), two key events take place that manifest conditions to gain insight into the two policies. (1) At simulation time $\tau(S_t)$ = 7.6 minutes, the first of two HVTs arrives just south of the TAI at approximate coordinates (-15, 10) and (2) a second HVT arrives at approximately $\tau(S_t)$ = 12.8 minutes to coordinates (-35, 25).



(a) $X_0^{\pi^{DROP}}$

(b) $X_0^{\pi^{LSTD}}$

(c) DROP Decision after HVT 32 arrives

(d) LSTD Decision after HVT 32 arrives

(e) Chooses HPT 33 over HPT 28

(f) Passes 36 and 33 to reach HVT 34

(g) Chooses HPT 36 over NAI 4

(h) Collects 36, 33, 25 and 4 on return to exit

Figure 16. A3P Problem Instance 1 (CRN seed 122) - policy contrast

The π^{DROP} ultimately experiences 23 epochs whereas the π^{LSTD} experiences 21. This investigation reveals several observations and possible insights. The higher density of initial targets seems to pull both policies initially to the southeast with a seemingly identical first decision see Figures 16a and 16b. After the arrival of the first HVT at 7.8 minutes into the mission, the π^{DROP} reroutes to include now-arrived HVT 32; however, π^{DROP} still decides to spend early playtime to service HPT 27. This decision ultimately seems to cost π^{DROP} the opportunity to service the not-yet-arrived HVT 34. In contrast, π^{LSTD} recognizes the arrival of HVT 32 and routes toward HVT 32 (and the TAI) with a more efficient route, choosing not to service HPT 27. This leaves π^{LSTD} enough playtime remaining to service the HVT 34 arriving at 12.8 minutes into the mission. In addition, it is interesting to note that π^{LSTD} passes HPT 36 en route to HVT 34 but later services HPT 36 on its return to the deep zone exit point.

We infer the following insights as a result of this case study. The π^{DROP} very well finds the optimal route for a given state. However, its lack of anticipation suggests it may be burning a portion of early playtime at epoch t that would have been better saved for a possible future arrival in epoch $t+1...t+2....$ The π^{LSTD} is a better steward of the playtime remaining ρ_t for any given epoch t since it preserved playtime in anticipation of future arrivals. We suppose this preferred use of playtime is attributed to the basis functions, which include the features expressed in Equations 14a and 14b. The π^{DROP} considers ρ_t as a constraint in Equation 27 but the DROP does not consider the amount of playtime it might use for its next decision relative to its current playtime remaining. This hypothesis is reinforced in the π^{LSTD} decision to closely bypass HPT 36 but then later service HPT 36. We supposed the π^{LSTD} basis functions, which include distance to the TAI centroid (Equation 12a), may be influential in these types of decisions.

The simulation for this problem alone instance took π^{DROP} approximately 30 minutes to compute. The simulation took π^{LSTD} just under 2 minutes to compute. This again reinforces our earlier data on the disparity in computational effort between the two policies.

We see further evidence of the π^{DROP} inefficient use of the playtime resource within Problem Instance 1 (see Figure 17). In this decision, π^{DROP} chooses a route that crosses over itself to collect the maximum rewards possible for epoch t. While this route is optimal for a specific t this criss-cross behavior is inefficient. The wasted playtime used to negotiate this optimal route could have been allocated toward flying deeper into the AO in anticipation of targets yet to arrive to the HPTL.

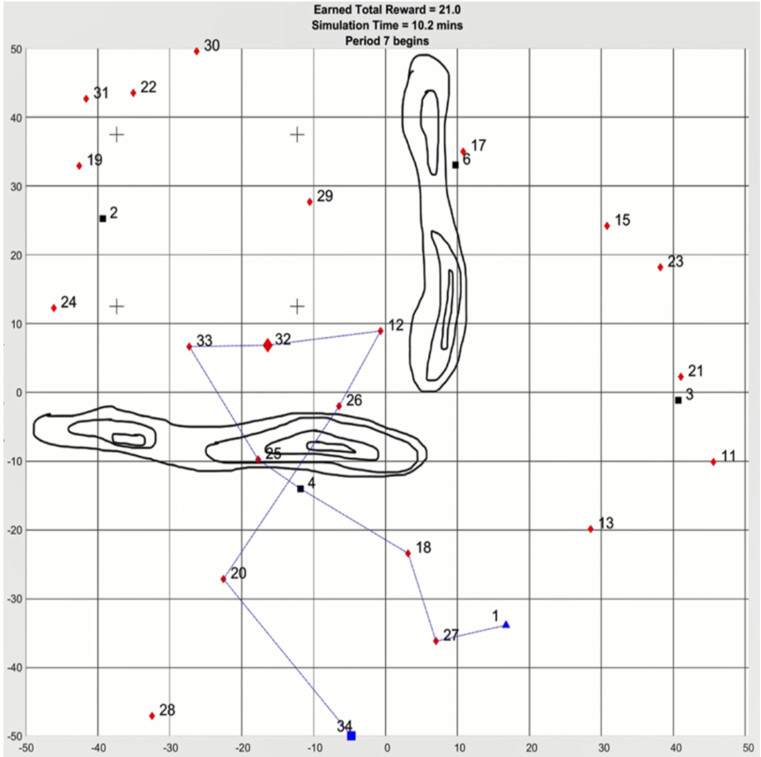


Figure 17. π^{DROP} Problem Instance 1 Misappropriation of ρ_t

From this experimentation we learned about the differing uses of playtime and reinforced the disparity in computational effort. It is reasonable to compare the amount of computational effort it takes to train and employ one π^{LSTD} to employing the π^{DROP}. Often, training and employing a π^{LSTD} is many times faster than employing the π^{DROP}. Indeed, the time disparity is so significant that it is appropriate to consider training an entirely new LSTD policy through Algorithm 2 and employing the brand new policy before the DROP finds a solution.

4.6 Additional case study analysis and insights

We follow a similar process of investigation into Problem Instances 2, 3, 4 and 5. The disparity in performance quality is again a result of the interaction of primarily $\frac{\lambda}{10}$ and ρ^{max}. If a problem instance has a higher ρ^{max}, the π^{DROP} attains a higher performance quality. However, as quality increases, computational effort also increases. The battlespace shape problem factor has a minor influence over quality, but less so than the other two problem factors.

Additional case study inspections confirm the π^{DROP} achieves a high quality because its behavior is optimal for any one system state (assuming no new target arrivals). When presented with a system state, there exists no better route than the route the π^{DROP} chooses. Any inefficiencies in route selection for a given system state that another decision rule may choose, such as π^{LSTD}, may be very costly when there are more opportunities to make inefficient decisions. A3P problem instances, where there are more opportunities to make decisions, occur in problems with increased target arrival rates and playtime.

The results of this campaign of experimentation equip us to glean several insights. We offer the following comments on π^{LSTD} performance in terms of quality, computational effort, and robustness relative to the benchmark π^{DROP}.

4.6.1 Performance - Quality

We investigate eight instances of the A3P over 30 runs with a 90% confidence (see Table 9). The π^{LSTD} outperforms π^{DROP} in one instance (Instance 1), and the π^{DROP} outperforms the π^{LSTD} in two instances (Instance 4 and 8). The two policies achieve parity in five instances (Instances 2, 3, 5, 6, 7). When only examining practical mean results the π^{LSTD} outperforms π^{DROP} in two of the eight instances and under performs in six of the eight.

The results suggest the π^{DROP} relative quality over π^{LSTD} is greater with increased arrivals and playtime. Conversely, the π^{LSTD} outperforms the π^{DROP} on instances of low target arrival and low playtime. Our case study analysis confirms the π^{DROP} makes optimal decisions for any one state considering only the current HPTL at epoch t. When the π^{DROP} is presented with sequential system states, its behavior, although myopic, continues to achieve a relatively high performance quality.

There could be several reasons for the disparity in performance quality between the two policies. First, the VFA is a linear model that must predict the value of being in a future state accurately to prove useful (recall Equation 10). The basis function evaluations that serve as the independent variables of the linear model must accurately and collectively capture the information necessary to facilitate discrimination of value among feasible future states. It is possible the linear value function approximation (VFA) does not capture the information necessary to make high-quality decisions. If the VFA poorly models the actual but unknown value function, performance suffers.

Our preliminary testing suggests the A3P's actual value function is likely non-linear, as all of our thirteen basis functions are second and third order interactions of the fundamental reward-based, spatial-based, and resource-based features. Indeed, all of the thirteen basis functions consider resource-based, playtime-centric features - a realization we only discovered near the climax of preliminary testing that resulted in a jump in performance quality. Another VFA approach such as support vector regression or neural network regression may be able to capture important features at the expense of clearly understanding these these features. Moreover, possibly the addition or removal of basis functions could greatly affect the accuracy of the VFA. For example, if we remove all features related to the type of node on which the AUCAV resides, the ADP policy would make decisions blind to the type of node as it relates to the contribution (i.e., reward) it just received for making a decision. Preliminary testing outlined in Figure 12 suggests the removal of such features is detrimental, and for that reason, we keep such features in our basis functions. Moreover, if a basis function were to be added that considers a route-based approach similar to the π^{DROP}, we could reasonably expect an increase in performance.

Other possible limitations to the performance quality of the π^{LSTD} are the ranges of the algorithmic parameter values we investigate (recall Table 6). It is possible that a combination of algorithmic parameter values we did not investigate using the same linear VFA, could return increased performance quality. We based our experimentation on the outcomes of many preliminary tests; however, there is no evidence to suggest that more testing with the same VFA cannot lead to increased performance quality.

Further testing should devote considerable attention to the elements of model construction related to how a policy considers the playtime resource, because the two policies' performance disparity seems to be centered around their allocation of playtime. The π^{DROP} arguably uses playtime perfectly but in a myopic manner only considering its current system state. The π^{LSTD} allocates playtime while considering anticipated but unknown targets. Possibly the inclusion of basis functions we outline in Chapter III may be included into a more powerful, non-linear, VFA to return higher-quality policies that will anticipate.

Additionally, our ADP solution approach is an offline method. After we finish training the π^{LSTD}, its VFA coefficients are fixed and do not change when solving the A3P. The results in Table 10 suggest the computational effort the π^{LSTD} requires to be trained *and* solve the A3P is less than the computational effort the π^{DROP} requires to solve. An online solution approach to solving the A3P is a reasonable endeavor that may increase performance quality.

Finally, any permutation of the suggestions above may yield an increase in ADP solution approach performance. We expect a JFC to prefer an autonomous attack aircraft equipped with a π^{LSTD} when the mission duration is short and the density of new arrivals is sparse. In contrast, if the density of new arrivals is high and mission duration is also high, the quality of the π^{DROP} likely makes it the preferred policy. We assume a notional JFC's intent in establishing the notional A3P instances to investigate and compare policies for the purposes of this research. If a JFC decides to employ an autonomous agent, we encourage reserving pre-combat check resources to rehearse what behaviors they expect an AUCAV to perform for a given set of rewards. Such pre-combat checks provide the opportunity for combat leaders to conduct final tuning of AUCAV behavior before likely enemy contact. This should support expectation management, leader understanding, and ideally, combat success.

4.6.2 Performance - Speed (computational effort)

If performance computational effort is influential in decision-making, it is reasonable to expect a JFC to prefer an autonomous aircraft equipped with a π^{LSTD} over the π^{DROP} for most problem instances. We suppose this is the case as the ADP policy's computational effort seems to be a fraction of the repeated solving of the orienteering problem within the problem instances we investigate (recall Table 10). Similar to performance quality, we encourage future employment to add computational effort expectations to practice pre-combat checks since a lagging or seemingly indecisive AUCAV could put friendly forces at risk.

4.6.3 Performance - Robustness

Through the lens of performance robustness, the π^{LSTD} under-performs the benchmark on problem instances with high target arrival rate and playtime. For instances with high target arrival rate and low playtime or low target arrival rate and high playtime, the π^{LSTD} and π^{DROP} qualities seem comparable. Given this, we suppose the π^{LSTD} is neither robust nor constrained to parochial employment for the problem instances we investigate. Indeed, the ADP policies computational effort may make it the only feasible policy. The π^{LSTD} has shown to be superior in the time-sensitive, short duration missions we investigate where the risk of success or failure is potentially heavier than the aggregation of many decisions over a longer, higher arrival battle. This superior performance of π^{LSTD} quality and speed seems robust for linear and non-linear battlespaces.

V. Conclusions and Future Research

This thesis examines the autonomous attack aviation problem (A3P). The intent of this research is to determine suitable policies that provide high-quality solutions to the A3P. We leverage a Markov decision process (MDP) framework to construct a mathematical model of the A3P. We then construct an approximate dynamic programming (ADP) (i.e., reinforcement learning) algorithm to generate decision rules (i.e., policies) that solve the MDP model. We define performance with respect to quality, computational effort, and robustness. We measure performance of our ADP solution approach as compared to a benchmark policy, the deterministic repeating orienteering problem (DROP) solution approach.

We introduce a baseline, realistic, problem instance of the A3P as a large-scale, near-peer, strike coordination and reconnaissance (SCAR) mission fought in a notional US Army division's deep area. Through extensive preliminary testing, we develop a set of basis functions to approximate the value function for our least squares temporal difference approximate policy iteration ADP solution approach. We then introduce eight notional problem instances of the A3P to provide a realistic venue to study the performance of our ADP solution approach relative to the benchmark policy.

We conduct computational experimentation to determine high-quality policies and evaluate policy effectiveness to study the efficacy of our proposed ADP solution approach. Finally, we leverage common random numbers to re-create the tactical situations and specific decision points to study the causes behind the disparity in the sequential decision-making under uncertainty performed by our ADP policy and benchmark policy.

5.1 Conclusions

The results of our investigations into the eight problem instances suggest the API-LSTD solution approach outperforms the benchmark on problem instances where AUCAV playtime is more constrained and new enemy arrivals are sparse. As arrival rate and playtime increase, the benchmark seems to perform with superior solution quality. The ADP solution approach is superior to the benchmark in terms of computational effort. Moreover, the ADP solution approach seems robust to changes in battle space shape.

This research is of interest to organizations concerned with the performance of autonomous military assets. Some organizations may be interested to know how autonomous aircraft may behave and adapt to develop new tactics, techniques, and procedures. Moreover, the acquisition and simulation community may be interested in the modeling of the A3P for possible implementation in training and wargaming. Throughout this research, we highlight the interdependence of human and machine decision-making. This blended decision-making seems realistic and increasingly prevalent in modern problem solving. Specifically, we outlined the roles and responsibilities of the Joint Forces Commander and their staff with emphasis on the inclusion of human intelligence officers' prediction of enemy action into AUCAV behavior (i.e., basis functions). Potentially the group of people most interested in this research are pilots themselves. Pilots may see AUCAVs successfully conducting deep SCAR missions as an alternative to a human completing the same mission. Moreover, a human pilot may be interested in the decision-making of an autonomous aircraft in the event they fly with an AUCAV as a wing-man for a combat mission.

5.2 Future Work

The crux of this research is to codify how an autonomous aircraft may perform when assigned an air-to-ground attack mission. Since we are one of the first research efforts to model the A3P in this way, we encourage future work in this area. We recommend three areas of deeper solution procedure research and three areas of problem instance research.

5.2.1 Future Work - Solution Procedure

Our efforts outlined in this thesis are subject to the amount of time available to conduct research. We have suggestions for interested parties whose intent is to pursue similar research efforts. Consider increasing the levels of algorithmic parameter levels and the span of the levels investigated. Computational experiments take time, and it is possible that continued computational experimentation of algorithmic parameter settings for π^{LSTD} may attain superior quality and robustness over the π^{DROP} in turn, thus providing more insights.

We recommend investigating new ADP policies that include the blending of the π^{DROP} approach to route planning with the π^{LSTD} anticipation. This may be approached by building a basis function which returns the expected total discounted reward based on a complete route if a certain node is chosen. To combat the costly computational effort, we suggest a roll-out algorithm that makes a truncated route, constrained by a finite number of steps less than the playtime remaining. If multiple AUCAVs are considered, potentially the blending of solution approaches could be investigated by allowing some AUCAVs to make decisions using the π^{DROP}, while others use the π^{LSTD}.

We recommend investigating neural network regression as a substitute to the least squares temporal difference approach to construct the value function approximation.

It is possible that a neural network ADP approach is more suitable to capture the complex, non-linear relationships necessary to increase performance quality. However, with neural network regression comes the possibility that any uncovered effective relationships of features may be too complex for human intuition to effectively understand.

5.2.2 Future Work - Problem Model

In addition to continued solution procedure research, we suggest further research into the modeling of the A3P itself. We encourage investigation into problem instances with the inclusion of multiple AUCAVs and multiple exit locations. Our recommendation's purpose is to increase relevancy and application of the A3P. We make a limiting assumption that only one AUCAV performs one mission at a time and that AUCAV cannot be destroyed. This assumption facilitates our initial research but does not comport with the reality that an adversary will desire and allocate resources to destroy the AUCAV. We suggest reviewing research wherein multiple autonomous agents work collectively toward a common goal such as Bertsekas (2021) and Bhattacharya *et al.* (2020). The consideration of multiple AUCAVs invites opportunity to simulate AUCAV attrition during combat. Moreover, the addition of exit locations is reasonable since there is current research into air refueling of helicopters.

We recommend instances with multiple areas of asymmetric enemy arrivals. In our work, we assume one location of increased density of enemy arrivals behind the L-shaped ridge line to the northwest. In more realistic scenarios, it's reasonable to foresee missions where there are multiple areas of increased enemy arrival density. We encourage this effort as it may reveal insights that would not otherwise be discovered.

Finally, we recommend investigating problem instances where the array of TAIs is not aligned with the actual increased density of enemy arrivals. Indeed, this may invite the inclusion of game theory into the A3P. A misalignment of the TAI to actual enemy arrivals simulates poor intelligence, which is something an adversary desires. Any consideration of a sentient adversary may also bring into question the predictability of AUCAV behavior. We suggest that any researcher concerned with application consider a fourth facet to solution performance possibly titled: solution predictability. As the behavior of an AUCAV becomes more predictable to an adversary, it may be less desirable to employ by practitioner of armed conflict.

In conclusion, we intend for the the data and insights presented in this research to support the development of viable AUCAV air-to-ground employment in future military operations.

Bibliography

Balas, Egon. 1989. The prize collecting traveling salesman problem. *Networks*, **19**(6), 621–636.

Banks, Jerry, Carson, John S, Nelson, Barry L, & Nicol, David M. 2013. *Discrete-Event System Simulation: Pearson New International Edition.* Pearson Higher Ed.

Barr, Richard S, Golden, Bruce L, Kelly, James P, Resende, Mauricio GC, & Stewart, William R. 1995. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, **1**(1), 9–32.

Bertsekas, Dimitri. 2021. Multiagent reinforcement learning: Rollout and policy iteration. *IEEE/CAA Journal of Automatica Sinica*, **8**(2), 249–272.

Bhattacharya, Sushmita, Badyal, Sahil, Wheeler, Thomas, Gil, Stephanie, & Bertsekas, Dimitri. 2020. Reinforcement learning for pomdp: partitioned rollout and policy iteration with application to autonomous sequential repair problems. *IEEE Robotics and Automation Letters*, **5**(3), 3967–3974.

Blumer, Anselm, Ehrenfeucht, Andrzej, Haussler, David, & Warmuth, Manfred K. 1987. Occam's razor. *Information Processing Letters*, **24**(6), 377–380.

Bradtke, Steven J, & Barto, Andrew G. 1996. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, **22**(1-3), 33–57.

Burket, Dennis S. 2019. *Large-scale Combat Operations: The Division Fight.* Army University Press.

Chao, I-Ming, Golden, Bruce L, & Wasil, Edward A. 1996. The team orienteering problem. *European Journal of Operational Research*, **88**(3), 464–474.

Chen, Xia, & Liu, Yongtai. 2019. Cooperative Task Assignment For Multi-UAV Attack Mobile Targets. *Pages 2151–2156 of: 2019 Chinese Automation Congress (CAC).* IEEE.

Collins, Liam. 2020. Affordable, Abundant, and Autonomous: The Future of Ground Warfare. *War on the Rocks*, 1–9.

Coryell, Brent, & Heap, Shayne. 2016. *CALL Newsletter - Decisive Action Training Environment at the National Training Center.* Tech. rept. US Army Center for Army Lessons Learned.

Coutinho, Walton Pereira, Battarra, Maria, & Fliege, Jörg. 2018. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, **120**, 116–128.

Dantzig, George B, & Ramser, John H. 1959. The truck dispatching problem. *Management science*, **6**(1), 80–91.

De Weerdt, Mathijs, & Clement, Brad. 2009. Introduction to planning in multiagent systems. *Multiagent and Grid Systems*, **5**(4), 345–355.

Department of Defense. 2016. *Joint Operating Environment 2035. The Joint Force in a Contested and Disordered World*. Washington, D.C.

Department of Defense. 2017. *Directive 3000.09: Autonomy in Weapon Systems*. Washington, D.C.

Department of Defense. 2018. *National Defense Strategy*. Washington, D.C.

Department of Defense. 2019a. *Joint Publication 3-03: Joint Interdiction*. Washington, D.C.

Department of Defense. 2019b. *Joint Publication 3-09: Joint Fire Support*. Washington, D.C.

Department of Defense. 2019c. *Joint Publication 3-30: Joint Air Operations*. Washington, D.C.

Department of Defense. 2020. *Joint Terms*. Washington, D.C.

Department of the Army. 2007. *Attack Recon Helicopter Operations*. Vol. Field Manual 3-04.126. Washington, D.C.

Department of the Army. 2011a. *Training Circular 7.100.2 Opposing Force Tactics*. Washington, D.C.

Department of the Army. 2011b. *Training Circular 7.100.4 Hybrid Threat Force Structure*. Washington, D.C.

Department of the Army. 2014. *Division Operations*. Vol. Army Techniques Publication 3-91. Washington, D.C.

Department of the Army. 2015. *Targeting*. Vol. Army Techniques Publication 3-60. Washington, D.C.

Department of the Army. 2016. *Deep Operations*. Vol. Army Techniques Publication 3-94.2. Washington, D.C.

Department of the Army. 2018. *Operational Terms*. Vol. Field Manual 1-02.1. Washington, D.C.

Department of the Army. 2019a. *Army Techniques Publication 2-01.3 Intelligence Preparation of the Battlefield*. Washington, D.C.

Department of the Army. 2019b. *Operations*. Vol. Army Doctrine Publication 3-0. Washington, D.C.

Dubins, Lester E. 1957. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, **79**(3), 497–516.

Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, *et al.* 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Pages 226–231 of: Kdd*, vol. 96.

Flood, Merrill M. 1956. The traveling-salesman problem. *Operations Research*, **4**(1), 61–75.

Freedberg, Sydney. 2020 (September). *A Slew To A Kill: Project Convergence*. [Online; posted 16-September-2020].

Freedberg, Sydney. 2021 (January). *Defiant X: Sikorsky, Boeing unveil FLRAA Deisgn*. [Online: posted 25-January-2021].

Frew, Eric, & Lawrence, Dale. 2005. Cooperative stand-off tracking of moving targets by a team of autonomous aircraft. *Page 6363 of: AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Gendreau, Michel, Guertin, Francois, Potvin, Jean-Yves, & Taillard, Eric. 1999. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science*, **33**(4), 381–390.

General Charles Q. Brown, Jr. 2020. *Accellerate Change Or Lose*. Tech. rept. United States Air Force.

Gordon IV, John, Mikolic-Torreira, Igor, Barnett, D Sean, Best, Katharina L, Boston, Scott, Madden, Dan, Tarraf, Danielle C, & Willcox, Jordan. 2019. *Army Fires Capabilities for 2025 and Beyond*. Tech. rept. RAND ARROYO CENTER SANTA MONICA CA SANTA MONICA United States.

Gülpınar, Nalan, Çanakoğlu, Ethem, & Branke, Juergen. 2018. Heuristics for the stochastic dynamic task-resource allocation problem with retry opportunities. *European Journal of Operational Research*, **266**(1), 291–303.

Gunawan, Aldy, Lau, Hoong Chuin, & Vansteenwegen, Pieter. 2016. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, **255**(2), 315–332.

Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics New York.

Jenkins, Phillip R, Robbins, Matthew J, & Lunday, Brian J. 2021. Approximate dynamic programming for military medical evacuation dispatching policies. *INFORMS Journal on Computing*, **33**(1), 2–26.

Jiang, Hao, & Liang, Yueqian. 2018. Online path planning of autonomous UAVs for bearing-only standoff multi-target following in threat environment. *IEEE Access*, **6**, 22531–22544.

Judson, Jen. 2019a (September). *Jumping into algorithmic warfare: US Army aviation tightens kill chain with networked architecture.* [Online; posted 05-September-2019].

Judson, Jen. 2019b (April). *US Army plans to field a future long-range assault helicopter by 2030.* [Online; posted 4-April-2019].

Judson, Jen. 2020a (December). *Army long-range cannon gets direct hit on target 43 miles away.* [Online; posted 21-December-2020].

Judson, Jen. 2020b (March). *Lockheed and Bell will compete head-to-head to build US Army's future attack recon aircraft.* [Online: posted 25-March-2020].

Kantor, Marisa G, & Rosenwein, Moshe B. 1992. The orienteering problem with time windows. *Journal of the Operational Research Society*, **43**(6), 629–635.

Kem, Jack D. 2018. Deep Maneuver: Historical Case Studies of Maneuver in Large-Scale Combat Operations. *Military Review*, **98**(5), 39.

King, Samuel Jr. 2020 (August). *Jolly Green II continues tests, completes first aerial refueling.* [Online; posted 10-August-2020].

Li, Zhenming, Li, Chaoyong, Zhuo, Shuo, & Chen, Sai. 2017. Uavs cooperative attack based on archimedes spiral. *Pages 1094–1098 of: 2017 17th International Conference on Control, Automation and Systems (ICCAS).* IEEE.

Mattis, Jim. 2018. *Summary of the 2018 National Defense Strategy of the United States of America.* Tech. rept. Department of Defense Washington United States.

Medeiros, André César, & Urrutia, Sebastián. 2010. Discrete optimization methods to determine trajectories for Dubins' vehicles. *Electronic Notes in Discrete Mathematics*, **36**, 17–24.

Naegele, Tobias. 2020. Why the U.S. Is Driving Allies to Buy Chinese UAVs. *Air Force Magazine*, 1–9.

Ning, Qian, Tao, Guiping, Chen, Bingcai, Lei, Yinjie, Yan, Hua, & Zhao, Chengping. 2019. Multi-UAVs trajectory and mission cooperative planning based on the Markov model. *Physical Communication*, **35**, 100717.

Pasztor, Andy. 2021. Forget Self-Driving Cars—the Pentagon Wants Autonomous Ships, Choppers and Jets. *The Wall Street Journal*, 1–2.

Pellerin, Cheryl. 2015. Work: Human-Machine Teaming Represents Defense Technology Future. *DoD News*, 1–6.

Pillac, Victor, Gendreau, Michel, Guéret, Christelle, & Medaglia, Andrés L. 2013. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, **225**(1), 1–11.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Vol. 842. John Wiley & Sons.

Psaraftis, Harilaos N, Wen, Min, & Kontovas, Christos A. 2016. Dynamic vehicle routing problems: Three decades and counting. *Networks*, **67**(1), 3–31.

Rettke, Aaron J, Robbins, Matthew J, & Lunday, Brian J. 2016. Approximate dynamic programming for the dispatch of military medical evacuation assets. *European Journal of Operational Research*, **254**(3), 824–839.

Schubert, Erich, Sander, Jörg, Ester, Martin, Kriegel, Hans Peter, & Xu, Xiaowei. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, **42**(3), 1–21.

Shima, Tal, & Rasmussen, Steven. 2009. *UAV Cooperative Decision and Control: Challenges and Practical Approaches*. Vol. 1. Siam.

Smithsonian. 2015. Actual Footage of Desert Storm's First Apache Strikes. *YouTube url, https://www.youtube.com/watch?v=RhpgCaPoBaE*.

Sutton, Richard S, & Barto, Andrew G. 2018. *Reinforcement Learning: An Introduction*. MIT press.

Trends, Global. 2017. Paradox of Progress. *A publication of National Intelligence Council*.

Tressel, Ashley. 2020. Army wants to modernize Ft. Irwin for future fight. *Inside Defense*, 1–2.

Ulmer, Marlin W, Goodson, Justin C, Mattfeld, Dirk C, & Thomas, Barrett W. 2017. *Dynamic vehicle routing: literature review and modeling framework*. Tech. rept. Working Paper, Technical University Braunschweig, Braunschweig.

Ulmer, Marlin W, Goodson, Justin C, Mattfeld, Dirk C, & Thomas, Barrett W. 2020. On Modeling Stochastic Dynamic Vehicle Routing Problems. *EURO Journal on Transportation and Logistics*, 100008.

Ulmer, Marlin Wolf. 2017. *Approximate Dynamic Programming for Dynamic Vehicle Routing*. Vol. 61. Springer.

USGS. 2021 (January). *USGS 7.5 min map of Fort Irwin, CA*. [Online; posted 31-August-2018].

Valavanis, Kimon P, & Vachtsevanos, George J. 2015. *Handbook of Unmanned Aerial Vehicles*. Vol. 1. Springer.

Vansteenwegen, Pieter, Souffriau, Wouter, & Van Oudheusden, Dirk. 2011. The orienteering problem: A survey. *European Journal of Operational Research*, **209**(1), 1–10.

Widyotriatmo, Augie, & Hong, Keum-Shik. 2008. Decision making framework for autonomous vehicle navigation. *Pages 1002–1007 of: 2008 SICE Annual Conference*. IEEE.

Yang, Fan, & Chakraborty, Nilanjan. 2019. Multirobot Simultaneous Path Planning and Task Assignment on Graphs with Stochastic Costs. *Pages 86–88 of: 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE.

Zhang, Yu, Chen, Jing, & Shen, Lincheng. 2012. Hybrid hierarchical trajectory planning for a fixed-wing UCAV performing air-to-surface multi-target attack. *Journal of Systems Engineering and Electronics*, **23**(4), 536–552.

Zhu, Rong, Sun, Dong, & Zhou, Zhaoying. 2005. Cooperation strategy of unmanned air vehicles for multitarget interception. *Journal of Guidance, Control, and Dynamics*, **28**(5), 1068–1072.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)* 25–03–2021	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* August 2019 — March 2021

4. TITLE AND SUBTITLE The Autonomous Attack Aviation Problem	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Goodwill, John C., MAJ, US Army	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENS-MS-M-162
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Mr. David M. Panson Strategic Development Planning Experimentation (SDPE) Office 1864 4th Street Wright-Patterson AFB, OH 45433 (937) 904-6539	**10. SPONSOR/MONITOR'S ACRONYM(S)** SDPE **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A. Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT An autonomous unmanned combat aerial vehicle (AUCAV) performing an air-to-ground attack mission must make sequential targeting and routing decisions under uncertainty. We formulate a Markov decision process model of this autonomous attack aviation problem (A3P) and solve it using an approximate dynamic programming (ADP) approach. We develop an approximate policy iteration algorithm that implements a least squares temporal difference learning mechanism to solve the A3P. Basis functions are developed and tested for application within the ADP algorithm. The ADP policy is compared to a benchmark policy, the DROP policy, which is determined by repeatedly solving a deterministic orienteering problem as the system evolves. Designed computational experiments of eight problem instances are conducted to compare the two policies with respect to their quality of solution, computational efficiency, and robustness. The ADP policy is superior in 2 of 8 problem instances – those instances with less AUCAV fuel and a low target arrival rate – whereas the DROP policy is superior in 6 of 8 problem instances. The ADP policy outperforms the DROP policy with respect to computational efficiency in all problem instances.

15. SUBJECT TERMS

approximate dynamic programming, reinforcement learning, artificial intelligence, autonomous attack aviation, targeting, deep attack

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Matthew J. Robbins, AFIT/ENS
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U	UU	96	(937)255-3636, x4606; matthew.robbins@afit.edu