3-2022

Approximate Dynamic Programming for an Unmanned Aerial Vehicle Routing Problem with Obstacles and Stochastic Target Arrivals

Kassie M. Gurnell

APPROXIMATE DYNAMIC PROGRAMMING FOR AN UNMANNED AERIAL VEHICLE ROUTING PROBLEM WITH OBSTACLES AND STOCHASTIC TARGET ARRIVALS

THESIS

Kassie M. Gurnell, Capt, USAF

AFIT-ENS-MS-22-M-134

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-22-M-134

APPROXIMATE DYNAMIC PROGRAMMING FOR AN UNMANNED AERIAL VEHICLE ROUTING PROBLEM WITH OBSTACLES AND STOCHASTIC TARGET ARRIVALS

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

Kassie M. Gurnell, B.S.O.R.

Capt, USAF

March 25, 2022

AFIT-ENS-MS-22-M-134

APPROXIMATE DYNAMIC PROGRAMMING FOR AN UNMANNED AERIAL VEHICLE ROUTING PROBLEM WITH OBSTACLES AND STOCHASTIC TARGET ARRIVALS

THESIS

Kassie M. Gurnell, B.S.O.R.
Capt, USAF

Committee Membership:

Dr. Matthew J. Robbins
Chair

Dr. Bruce A. Cox
Member

AFIT-ENS-MS-22-M-134

Abstract

The United States Air Force is investing in artificial intelligence (AI) to speed analysis in efforts to modernize the use of autonomous unmanned combat aerial vehicles (AUCAVs) in strike coordination and reconnaissance (SCAR) missions. This research examines an AUCAV's ability to execute target strikes and provide reconnaissance in a SCAR mission. An orienteering problem is formulated as an Markov decision process(MDP) model wherein a single AUCAV must optimize its target route to aid in eliminating time-sensitive targets and collect imagery of requested named areas of interest while evading surface-to-air missile (SAM) battery threats imposed as obstacles. The AUCAV adjusts its route depending on the arrival locations of the SAM batteries and targets into the battle-space. An approximate dynamic programming (ADP) solution approach is developed wherein mathematical programming techniques are utilized with a cost function approximate (CFA) policy to develop high quality AUCAV routing policies to improve SCAR mission performance. The CFA policy is compared to a deterministic repeated orienteering problem (DROP) benchmark policy across four instances that explores varied arrival behaviors of dynamic targets and SAM batteries. When the AUCAV is allotted 120 minutes to complete its mission and SAM batteries arrive into the battle-space, results show that the proposed CFA policies outperform the DROP policy. Overall, the proposed CFA policies perform nearly the same or better than the DROP policy in all four instances.

Key Words: Markov decision process (MDP), approximate dynamic programming (ADP), reinforcement learning (RL), artificial intelligence (AI), orienteering problem (OP), vehicle routing problem (VRP), targeting, cost function approximation (CFA), direct lookahead approximations (DLA), mesh adaptive direct search (MADS)

Acknowledgements

I thank my thesis advisor, Dr. Matthew Robbins, and reader, Dr. Bruce Cox for supporting me with this thesis. This work would not exist without my amazing parents, close family, and friends who encouraged me throughout this journey. Most importantly, I could not have completed this journey without God's provided wisdom, love, and grace. Thank you.

Kassie M. Gurnell

Table of Contents

Page

List of Figures

List of Tables

Table Page

APPROXIMATE DYNAMIC PROGRAMMING FOR AN UNMANNED AERIAL VEHICLE ROUTING PROBLEM WITH OBSTACLES AND STOCHASTIC TARGET ARRIVALS

I. Introduction

According to the United States Secretary of Defense (SecDef), the near future focus of the United States (US) military is to modernize current "capabilities to meet the advance threats of tomorrow" and ensure the US military remains the "world's preeminent fighting force" (Department of Defense, 2021). The SecDef's focus can be achieved by the US Department of Defense (DoD) effectively aligning its resources to evolving threats (Department of Defense, 2021). This thesis supports the DoD's top priorities for the future that involve using autonomous unmanned combat aerial vehicles (AUCAVs) for Suppression of Enemy Air Defense (SEAD) and strike missions. These priorities include advancement in artificial intelligence (AI), reconnaissance aircraft capabilities, Combatant Command (COCOM) stratagems, and deterring adversaries. By developing approximate dynamic programming (i.e., model-based reinforcement learning) algorithms for AUCAV path planning and target selection, we can explore the Air Force's capability to strike deep, time-sensitive targets and deter adversaries in direct alignment with the DoD's prime initiatives (Office of the Under Secretary of Defense (Comptroller)/Chief Financial Officer, 2021).

The Air Force is working to achieve technological advancement and modernization of the F-35 Joint Strike Fighter, F-15EX Eagle II, and Joint Direct Attack Munition (JDAM) high priority assets (Office of the Under Secretary of Defense (Comptroller)/Chief Financial Officer, 2021) through their SecDef approved programs

(Department of Defense, 2021). One aspect these assets have in common is they are all essential to COCOM missions and can work in conjunction with an AUCAV's ability to strike high value targets.

1.1 DoD Initiatives

There are multiple ways an AUCAV can be used to support COCOM missions. One unique way is to provide reconnaissance regarding the location of a time sensitive target (TST), enabling other allied aircraft or ground assets to strike the target. The fifth generation F-35 can engage ground targets, including surface-to-air missiles (SAMs), at long ranges without detection and use precision weapons to successfully complete air-to-ground missions (Military Advantage, 2014). AUCAVs may not be as effective against SAMs and may be shot down by them. However, AUCAVs have the ability to perform reconnaissance on requested named areas of interest (NAIs) or target types more fitting for other military assets to strike such as the F-35 or B-52.

The F-15EX is a SecDef approved program that, unlike the F-35, is not stealth and cannot go undetected behind enemy lines. However, the Air Force has considered pairing the F-15EX with a stealth fighter and using the pair as a long-range, air-to-air missile launch platform (Mizokami, 2021). Although the F-15EX is also capable of air-to-ground strikes, the aircraft's main strengths are its radar and ability to carry a large weapons payload, including two dozen air-to-air missiles or hypersonic weapons (Mizokami, 2021). This combat capability is important to consider because pairing a stealth aircraft (e.g., F-35) with an aircraft not capable of the same attribute (e.g., F-15EX) to complete a time-sensitive target strike mission as the result of an AUCAV's target confirmation capability may attain superior performance.

In each COCOM's area of responsibility (AOR), commanders request images of NAIs and high value target strikes. An AUCAV can service the commander's request,

assuming an absence of enemy threats capable of shooting down the AUCAV (e.g., SAM batteries). However, this assumption neglects the reality that an enemy can impose obstacles that can gravely affect well-planned missions. Path planning must incorporate intelligence information of no-fly zones (NFZ) for threat avoidance purposes. The approximate dynamic programming (ADP) algorithms discussed in this thesis will explore how unforeseen NFZs or combat zones (e.g., as a result of SAM batteries) can impact AUCAV target selection and how an AUCAV can learn to avoid those zones over time.

The US military has executed operational tests and evaluations (OT&E) of Unmanned Combat Aerial Vehicles (UCAV) using JDAMs to strike targets (Butler and Colarusso, 2002). As a result, it is assumed the AUCAVs in this thesis employ JDAMs to strike high value targets. The JDAM is capable of being individually directed to its target using in-flight target update (IFTU) messages transmitted from the Joint Surveillance Target Attack Radar System (JSTARS) (Butler and Colarusso, 2002). Tests have been completed showing significant improvements in using the Affordable Moving Surface Target Engagement (AMSTE) instead of the JSTAR resulting in a UCAV's increased capability to strike moving targets. This development should be further explored as a follow on but will not be discussed in detail in this thesis.

The US Special Operations Command (USSOCOM) is investing in artificial intelligence (AI) to speed analysis (Office of the Under Secretary of Defense (Comptroller)/Chief Financial Officer, 2021). This thesis focuses on establishing an AI algorithm that will enable combatant commands, like USSOCOM, to promptly and effectively execute target strikes and provide reconnaissance of requested NAIs. In addition to the COCOMs, Joint Intelligence Support Elements (JISE) and Joint Task Forces (JTFs) rely on reconnaissance aircraft due to their role in managing all forms of reconnaissance and surveillance of the enemy that are necessary for understanding the situation,

identifying objectives and opportune targets, and providing warning to forces (Department of Defense, 2018a). All three of these applicable military organizations can be provided a higher volume of intelligence information if the current AUCAV path planning AI algorithm is improved, resulting in more target strikes and images of NAIs given current limiting resources (e.g., fuel capacity, ammunition, or time in theatre).

1.2 Air Force Doctrine on Targeting

Targeting is a command function requiring commander oversight and involvement to ensure proper execution (Department of the United States Air Force, 2019). It is not the exclusive province of one type of specialty or division, such as intelligence or operations, but blends the expertise of many disciplines (Department of the United States Air Force, 2019). This thesis explores this blending of expertise by incorporating the intelligence received prior to AUCAV missions with US military joint, tactical, and Air Force doctrine. It is best to consider both joint doctrine and Air Force doctrine to better understand how the Air Force defines targets. According to joint doctrine, a target is an entity or object considered for possible engagement or other actions (Department of Defense, 2018b). Entities can be described as facilities, individuals, virtual (nontangible) things, equipment, or organizations (Department of the United States Air Force, 2019).

There are two categories of targeting: deliberate and dynamic (Department of the United States Air Force, 2019). Deliberate targeting applies when there is sufficient time to add the target to an air tasking order or other plan. Deliberate targeting includes targets planned for attack by on-call resources. Dynamic targeting includes targets that are either identified too late or not selected in time to be included in deliberate targeting, but when detected or located, meet criteria specific to achieving

objectives.

This thesis seeks to determine optimal AUCAV routes for selecting a combination of both deliberate and dynamic targets. The AUCAV enters the battle-space with a set of requested deliberate targets to strike or reconnoiter. Once in the battle-space, the AUCAV encounters new target requests (i.e., dynamic target arrivals) and must recalculate its optimal target selection route accounting for the new arrivals.

Two subsets of targets that require special consideration are sensitive and time-sensitive (Department of the United States Air Force, 2019). Sensitive targets are targets for which the commander has estimated that the physical and collateral effects on civilian and/or noncombatant persons, property, and environments occurring incidental to military operations exceed established national level notification thresholds (Department of Defense, 2018b). Sensitive targets are not always associated with collateral damage (Department of the United States Air Force, 2019). They may also include those targets that exceed national-level rules of engagement thresholds, or where the combatant commander determines the effects from striking the target may have adverse political ramifications (Department of the United States Air Force, 2019). Time-sensitive targets are joint force commander validated targets or sets of targets requiring immediate response because they are highly lucrative, fleeting targets of opportunity, or they pose (or will soon pose) a danger to friendly forces (Department of Defense, 2018b).

This thesis focuses on an AUCAV striking time-sensitive targets and providing reconnaissance on NAIs that may include sensitive targets while avoiding NFZs represented as SAM battery threat areas. This is accomplished by solving an unmanned aircraft orienteering problem with stochastic target arrivals while avoiding obstacles using ADP methods, integer programming techniques, and the Markov decision process (MDP) model framework. The vehicle routing problem MDP model framework

is leveraged for the baseline analysis of AUCAV target selection while avoiding obstacles (i.e., SAM batteries) and determining which time sensitive targets should be destroyed within an allotted time period. Then an ADP solution approach using a CFA policy is implemented to optimize the AUCAV target route, utilizing the predicted locations of future dynamic time sensitive target and obstacle arrivals when making decisions.

The remainder of this thesis is structured to address literary works similar to an autonomous vehicle orienteering problem with stochastic target arrivals in Chapter 2, the problem formulation framework and solution approach in Chapter 3, computational testing and results in Chapter 4, and the conclusion in Chapter 5. In detail, Chapter 2 explores similar path planning problems with stochastic arrivals, service times, and wait times from an ADP perspective. Chapter 3 gives insight into the methods used to model and solve the problem. Chapter 4 reveals analyzed results and recommendations. Chapter 5 concludes this thesis with future recommendations for producing improved solution procedures for AUCAV target selection and evasion of enemy threats.

II. Literature Review

This thesis extends the work of Maj Goodwill (2021) who focused on AUCAVs performing air-to-ground attack missions making sequential targeting and routing decisions under uncertainty. He referenced dynamic stochastic vehicle routing, task-resource allocation, orienteering, approximate dynamic programming (ADP), and unmanned aerial vehicle (UAV) problems. This thesis references similar topics with additional excursions. The orienteering problems, vehicle routing problems, and cost function approximation (CFA) algorithms are discussed at the beginning of this section. Next, this section describes the methods, techniques, and model frameworks used to solve the unmanned aircraft orienteering problem with stochastic target arrivals.

2.1 Literary Broadening

2.1.1 Orienteering

Shu et al. (2018) discuss a stochastic orienteering problem on a network of queues wherein a traveler decides to stay at the current queue or change queues. The problem is represented by a seven dimension state space summarized by the traveler's status, the historical record of locations (not) visited by the traveler, and the queue length at the specified time the traveler reneged on previously visited queues. This problem formulation can be applied to AUCAVs collecting reconnaissance and surveillance intelligence information at a target location where the AUCAV is waiting for a clear view of the target. The article recommends compound-one-step rollout policies be performed over prior solutions when solving similar problems.

Blum et al. (2007) discuss an orienteering problem and a discounted reward traveling salesman problem where the high value nodes must be reached in a timely

manner. The state space consists of the location of the salesman and a record of previously visited nodes to ensure the same node is not visited more than once. Multiple constant-factor algorithms are explored to achieve the most accurate results. A regular orienteering problem focuses on reaching as many high value nodes within a given distance. Instead of using distance as the limiting factor, this article uses an infinite horizon where the goal is to reach high value nodes in a timely manner. The reward functions generate a low reward for reaching a node late and a higher reward for reaching a node on time. This problem formulation could be used for an AUCAV providing ground support in a timely manner. However, the problem formulation must be modified to a stochastic orienteering problem where the nodes (i.e., air-to-ground support requests) would arrive according to some distribution to better represent military air to ground support requests resolved using AUCAVs.

Campbell et al. (2011) address an orienteering problem with stochastic travel and customer service times representing a company with high demand that occasionally exceeds its daily supply. The stochastic travel times in their work can be represented in this study as the AUCAV's deterministic travel time calculated based on its speed and the distance between its current location and future destination. The high demand referenced in their problem is represented in this research as the number of requested NAIs and time-sensitive targets the AUCAV must service to meet demand. Lastly, the AUCAV's limited payload of weapons and nearby supporting aircraft available to service targets in this thesis can be representative of the daily supply discussed in their work. The structure of the problem formulated by Campbell et al. (2011) is similar to the structure used in this thesis with the exception of the stochastic travel time and limited supply units available to meet demand. It is assumed an unlimited supply of supporting aircraft missiles are available. Therefore, freeing the AUCAV from the obligation of using its own missiles. It is also assumed

each target is serviced as soon as the AUCAV reaches it without variability in travel or missile delivery time. The stochasticity in the time it may take for the AUCAV or a supporting aircraft missile to launch then destroy a requested target is not addressed in this thesis.

The MDP formulations used in the three problems discussed above are referenced to obtain the static orienteering problem formulation used in this thesis. The minor differences between the problems discussed above and this thesis is that unlike Shu et al. (2018), this thesis does not incorporate a queue at each desired location. The AUCAV is assumed to be the first aircraft to reach a live target or NAI. Blum et al. (2007) places a time limit on the visitation of each individual node while this thesis does not impose a time limit on each node (or target). Also, unlike Campbell et al. (2011), the AUCAV is not limited by its own or other aircraft's supply of ammunition.

2.1.2 Vehicle Routing

Vehicle routing problems are at the center of this AUCAV target selection problem. The AUCAV is responsible for servicing as many time sensitive targets as possible in a limited amount of time. Ulmer et al. (2018) and Ulmer et al. (2020) discuss ADP and MDP model frameworks for commercial customer service vehicle routing problems.

Ulmer et al. (2018) develop an ADP algorithm that optimizes a vehicle route based on a number of customer service requests, the customer products available in the vehicle, and predicted customer service requests one time-period in the future. The goal is for the vehicle to reach as many customers as possible within one workday. The problem compares the vehicle's supply to the customer demand and throughout the day selects which customers will receive service using value function approximations (VFAs), ADP, and policy function approximation (PFA). Decisions to accept or

postpone a request are permanent and cannot be reversed in future decisions (Ulmer et al., 2018). For this thesis, the AUCAV will carry a limited supply of weapons but is assumed to have an unlimited supply of allied aircraft support which will be able to shoot targets reached by the AUCAV. Knowing this, the AUCAV must decide which target requests are accepted, postponed, or cancelled for the current mission. If accepted, the AUCAV travels to the target and the target is considered destroyed by the AUCAV or allied aircraft missiles once visited. If postponed, a previously accepted target remains accepted but the AUCAV's expected arrival time is delayed. If cancelled, a previously accepted target is removed from the AUCAV's route. The AUCAV has the freedom to cancel and re-accept targets interchangeably and fluidly. Unlike the approach offered by Ulmer et al. (2018), each target has a differing estimated payoff. Moreover, a target is not serviced based on a first-time, first-serve basis, but based on if it's a part of the optimal route that maximizes the overall cumulative payoff.

Ulmer et al. (2020) emphasize sophisticated stochastic dynamic vehicle routing problem (SDVRP) solution procedures using an MDP modeling framework. For their MDP formulation, the state space is a three-tuple including the vehicle location, the time of arrival to the vehicles current location, and a vector containing the service status of each customer. The action space is represented by customer assignments constrained by the vehicles' ability to make it back to the depot within a predetermined allotted amount of time. The reward function awards one unit of reward every time the vehicle services a customer. The objective is to maximize the cumulative expected reward by servicing many customers and returning to the depot within an allotted time period. The basic AUCAV target selection problem can be solved using the model frameworks discussed in this section. However, a lookahead approximation strategy can be implemented to explore how to achieve higher cumulative expected

rewards considering predicted locations of future targets.

2.2 Lookahead Approximations

Powell (2019) discusses the nuances of direct lookahead approximation (DLA) algorithms to solve a stochastic optimization problem. DLAs are considered a brute force approach used to solve a simplified model of the future. For state-independent ADP problems, there are single-period and multiperiod DLA policies.

Single period DLA policies considered for this thesis are knowledge gradient, expected improvement (EI), and Thompson sampling. Frazier et al. (2008) explain that the knowledge gradient policy maximizes the expected increment in the value of information in a single time period, where the value is measured according to a terminal utility function. The knowledge gradient (KG) is expressed through a formula that measures how much better we can find the best decision. The EI policy is similar to the knowledge gradient but is theoretically executed using an absolute value concept as it only focuses on the non-negative values (i.e., improvement) outputted by the same formula. The EI policy function outputs the maximum between the value zero and the KG function ensuring the result is always capturing a non-negative improvement. Thompson sampling works by sampling from a current distribution based on previous data, then a posterior distribution is generated through using the maximum likelihood function (Thompson, 1933).

One multiperiod DLA policy considered for this thesis is an extended variation of the KG policy. This policy considers when the experiment has noisy data. Instead of predicting a single period ahead, this policy executes multiple tests at the current system state then finds the optimal number of tests which results in the maximum average value of information.

For state-dependent ADP problems, DLA approximations can be implemented.

There are three DLA approximation strategies that are considered for this thesis. The first is a deterministic lookahead strategy that could be executed as a deterministic dynamic multiperiod linear programming problem. Solving the shortest path problem using a rollout policy is an example of the deterministic lookahead strategy. The rollout policy is a powerful strategy that interprets a search over a restricted set of policies over multiple time-periods. In addition to the shortest path problem, a time-dependent inventory problem can be solved using the rollout policy. This thesis takes into account the AUCAV's shortest paths to each target, maximum time allotted to complete its mission, fuel capacity, and unlimited support of allied missiles which incorporates both the time-dependent inventory and shortest path problem formulations. The second strategy is the stochastic lookahead procedure using a Monte Carlo tree search (MCTS) for discrete decisions. This strategy explores multiple decisions using the rollout policy and a backpropagation algorithm to update the value of each decision using a value function approximation formula. One important aspect of this strategy is it uses the principle of information relaxation to take a sample of the future and then solve the resulting deterministic problem assuming its ability to look into the future (Jiang et al., 2020). The third strategy is for stochastic lookahead models with vector-valued decisions implemented using a two-staged stochastic programming and decomposition algorithm. This strategy is broken out into three steps: 1) A decision is made at time t, 2) then future sample paths are created based on a sampled set of sample paths, and 3) making all remaining decisions based on the future sample paths created in the previous step.

2.3 Cost Function Approximation

Powell (2021) discusses the value of using a parametric CFA to solve very-large scale problems in Chapter 13 of his book. The general formulation of a parametric

CFA is represented by Equation 1 below.

$$\mathcal{X}^{CFA}(S_t|\theta) = \underset{x_t \in \mathcal{X}_t^{\pi}(\theta)}{\arg\max} \bar{C}^{\pi}(S_t, x_t|\theta), \tag{1}$$

wherein $\bar{C}^{\pi}(S_t, x_t|\theta)$ is a parametrically modified estimated cumulative reward as determined by policy π, which informs the value of the tunable parameter θ. Let x_t represent a decision chosen from a set of feasible decisions $\mathcal{X}_t$ at time t when the system environment is in state S_t. The $\mathcal{X}_t^{\pi}(\theta)$ indicates the feasible region determined by policy π with tunable parameters θ. Therefore, $\mathcal{X}^{CFA}(S_t|\theta)$ is a tunable policy in which a search for some policy θ is used to find the optimal estimated cumulative reward. The policy θ can be represented as a scalar or a vector. For this thesis, θ is represented as a vector.

This thesis uses an objective-modified CFA approach in which the objective function $\bar{C}^{\pi}(S_t, x_t|\theta) = C(S_t, x_t) + \Sigma_{f \in \mathcal{F}} \theta_f \phi_f(S_t, x_t)$ utilizes a linear cost function correction approach as represented by Equation 2.

$$\mathcal{X}^{CFA-cost}(S_t|\theta) = \underset{x_t \in \mathcal{X}_t}{\arg\max}(C(S_t, x_t) + \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t, x_t)), \tag{2}$$

The $C(S_t, x_t)$ component represents the immediate reward for choosing the next target or NAI (determined by decision x_t) given the current state of the system (or AOR environment) S_t. The $\Sigma_{f \in \mathcal{F}} \theta_f \phi_f(S_t, x_t)$ component represents some linear parametric function dependent on θ to calculate a numeric value. This numeric value is added to the immediate reward $C(S_t, x_t)$, thus acting as a corrector. In addition to the linear cost function correction approach, a dynamic shortest path problem CFA approach is examined because it further relates to the AUCAV path planning and vehicle routing components of the problem presented in this thesis. The dynamic shortest path problem objective function is represented by Equation 3.

$$\min_{\pi} F^{\pi}(\theta) = \mathbb{E}\{\sum_{t=0}^{T} \sum_{(i,j)\in\mathcal{N}} \hat{c}_{t+1,ij} X^{\pi}(S_t|\theta)|S_0\}, \tag{3}$$

wherein $F^{\pi}(\theta)$ is the distance traveled based on θ given policy π and $X^{\pi}(S_t|\theta)$ is an indicator variable that is 1 if it specifies that the traveler should travel via route (i, j) at time t, incurring the cost $\hat{c}_{t+1,ij}$ (Powell, 2021). The quantity $\hat{c}_{t+1,ij}$ is the cost of taking route (i, j) after deciding to at time t.

The linear cost function correction and the dynamic shortest path approaches are merged and transformed into a novel objective-modified CFA approach in this thesis which will be discussed further in Chapter 3.

There are three key actions involved with using a parametric CFA; (1) designing the parameterization, (2) evaluating a parametric CFA, and (3) tuning the parameters. To design the parameterization, a deterministic optimization model (i.e., a benchmark policy) must be initialized allowing for a parameterization to be chosen to improve upon the results of the initialized deterministic approximation. To evaluate a parametric CFA, it is best to simulate multiple problem instances over numerous applicable policies to help determine which policy on average yields the best results. Lastly, tuning the parameters entails formulating an applicable and effective objective function. For example, the linear cost function correction approach can be represented by a 1st order regression equation, 2nd order regression equation with interactions, and/or an indicator function.

2.4 Introduction of ORTHOMADS

The orthogonal mesh adaptive direct search (ORTHOMADS) algorithm aids in accomplishing two of the three key actions involved with using a parametric CFA. The algorithm allows us to choose and initialize the deterministic optimization model and evaluate the parametric CFA via simulation. ORTHOMADS performs a poll

and search sampling of input variable value(s), x, to evaluate solution, $f(x)$, and records the optimal solution, $f(x^*)$. This thesis will use θ as the input variable, seeking the best average estimated cumulative reward, $f(\theta)$ over multiple iterations. More detail is provided in Chapter 3. The traditional mesh adaptive direct search (MADS) algorithm is an iterative method that conducts a poll and search during each iteration where the objective function and a test for feasibility is evaluated finitely at many trial points (Abramson et al., 2009). The goal of each iteration is to generate a feasible solution that results in a smaller (i.e., improved) solution than the current best feasible solution.

2.5 Literature for Related Topics

2.5.1 AUCAV Behavior from a Gaming Perspective

Jansen et al. (2018) discuss an MDP problem formulated to model the PAC-MAN arcade game. Value iteration is used to determine the PAC-MAN's movements. Policy iteration is used to determine the benefit of using the shield during the game by reporting the difference in the maximum average reward when the PAC-MAN does or does not use a shield. The objective is to maximize average reward of the game score by avoiding adversaries and retrieving food to gain points. Results indicate that PAC-MAN achieves a higher score in both the small and classic PAC-MAN game when using the shield. The PAC-MAN problem applies to this thesis because it focuses on retrieving desired objects while actively evading an adversary.

The ideas behind the PAC-MAN problem can be explored further as it relates to applying a security shield for the AUCAV problem. This will allow additional exploration of a simple cyber defense protection system (or jamming capability). Zheng and Siami Namin (2018) reference optimizing network security using a moving target defense strategy and offers an example of how to create an MDP problem

formulation with cyber attacks. This idea would be good to explore in future theses.

Pang et al. (2019) use a finite horizon structured MDP problem to help depict a character's story line trajectory and find an optimal policy to winning the game StarCraft II. The state space is represented by the gamer's own global observation at a specific time in the game. Value and policy iteration are used to reveal the type of game decisions and gamer training required to maximize the probability of a player winning the game. This could be explored as an excursion for AUCAV or adversary behavior. Once the AUCAV and/or adversarial behaviors are understood, additional analysis can be done to determine the combat environments the AUCAV, adversary, or both entities can thrive in based on their predetermined behaviors.

Thue and Bulitko (2012) discuss how a Procedural Game Adaptation (PCA) MDP algorithm can be used to affect a player's game-play by changing the MDP transition probability matrix to better represent the player's habits. The goal is to maximize the reward of the game player by generating an MDP formulation of the video game and allowing updates in the MDP's transition probability matrix based on the game player's habits. The application of the PGA algorithm results in optimal policies for the game player to execute to achieve an optimal score. Similar to the StarCraft II MDP problem (Pang et al., 2019), the proposed algorithm can be used in future AUCAV problems where the AUCAV habits can be stored and used to create the optimal environment for its behavior.

2.5.2 Pursuit-Evasion

Ragi and Chong (2013) address a path-planning algorithm to guide unmanned aerial vehicles (UAVs) for tracking multiple ground targets based on the theory of partially observable Markov decision processes (POMDPs). The UAVs are tasked to track moving ground targets while evading threats and obstacles. The state space is

a four tuple consisting of the UAV's movement status and sensor state, the target status, and the UAV tracker status (Ragi and Chong, 2013). Insight gained from Ragi and Chong (2013) informs formulation of a state space for the problem being analyzed in this thesis.

Souidi et al. (2017) address a pursuit-evasion MDP through pursuing mobile agents in an uncertain environment while also avoiding obstacles. The pursuers decide to either pursue the evader or avoid an obstacle at each decision epoch. This decision depends on the state variable: sensor status line of sight, distance from the evader, and distance from the obstacle. The evaders are tasked with moving away from the pursuers at each time-step. A value iteration algorithm is used to determine an optimal policy. This thesis focuses on AUCAV target selection while avoiding potential SAM threats (i.e., obstacles). Souidi et al. (2017) provide insight on how to consider obstacles while optimizing pursuit or evasion routes.

Murali (2018) discusses an MDP model representing the cat and mouse predator-prey relationship of Tom (cat) and Jerry (mouse). Reinforcement learning is used to reveal optimal policies for Tom and Jerry over multiple scenarios to optimize their probability of winning in a two dimension grid world. Four cases were explored and analyzed to see the results of Jerry's probability of winning by evasion and Tom's probability of winning by capturing Jerry by game termination. At its simplest form, the AUCAV target selection problem with stochastic arrival, solved using an MDP process, has similar characteristics as the methods used in this predator-prey problem formulation.

The methods, techniques, and model frameworks discussed in this chapter are used to construct the methodology and problem formulation presented in the next chapter.

III. Methodology

3.1 Problem Definition

3.1.1 AUCAV Capabilities

The USAF employs UCAVs for close air support, strike coordination and reconnaissance (SCAR), and combat survival and recovery (CSAR) missions. One of the most popular UCAVs, the MQ-9, carries a 3,850 pound maximum payload of four Hellfire missiles and two GBU-12 Paveway II Laser-Guided Bombs or two 500-pound GBU-38 Joint Direct Attack Munitions (JDAM). The problem formulated in this thesis considers a notional, newly fielded AUCAV with a maximum 3,850-pound payload.

Karas (2017) provided an informative report on a CSAR capstone exercise executed on the Nevada Test and Training Range in 2017 using MQ-9 sensors and radio to coordinate with a survivor on the ground, nearby A-10s, and helicopter rescue forces to protect the survivor from enemy forces and facilitate their recovery. In a SCAR mission, an MQ-9 can offer situational awareness and clearance for other aircraft to fly into the area and fire at specific targets. Both SCAR and CSAR missions carry out similar agendas of using a UCAV to provide reconnaissance for other military assets to recover or destroy a desired time-sensitive target. This thesis considers a SCAR mission that allows the AUCAV to decide if it will destroy a confirmed target itself or refer the target to another military asset such as the F-35, F-15EX, or B-52 as discussed previously in Chapter 1.

The MQ-9 has an endurance of over 27 hours and is capable of completing a SCAR mission within the allotted time-frame. However, each time the AUCAV is deployed it is assumed to be deployed from a CONUS location to a requested OCONUS battlespace which requires an estimated travel time of 12 hours. This leaves an absolute maximum of 3 hours for the AUCAV to complete one SCAR mission. For safety

reasons, the maximum amount of time the AUCAV is allowed to complete its mission is 2.5 hours.

3.1.2 Benchmark Scenario

Figure 1 provides a simple visual example of the benchmark problem. Showing an entry and exit point for the AUCAV along with various time-sensitive targets, NAIs, and three obstacles representing the adversary's SAM battery striking range.

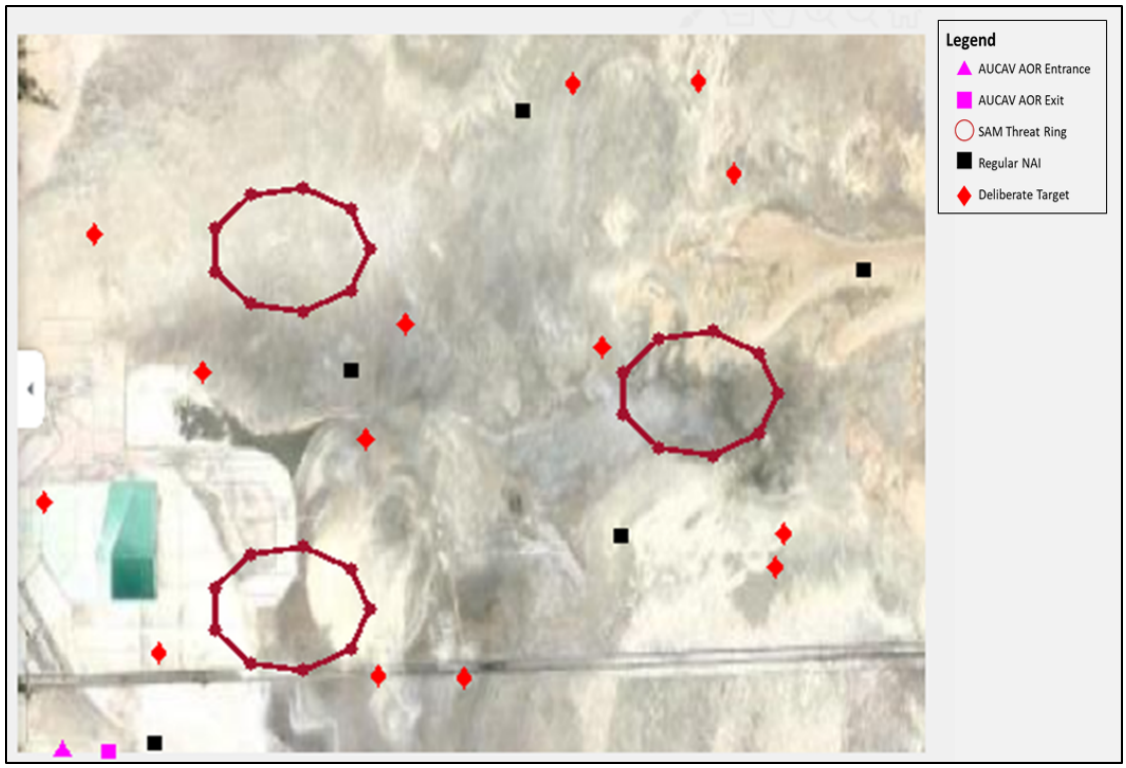

Figure 1: SCAR Benchmark Scenario

When a COCOM commander requests NAI imagery and target strikes prior to an AUCAV's entry into the desired AOR, the requested targets are identified as deliberate targets in which their locations are confirmed prior to the mission. In Figure 1, the red diamonds and black squares represent the deliberate targets and NAIs requested, respectively. The AUCAV is tasked with confirming and providing

imagery of the time-sensitive target locations as a means to help allied forces destroy targets. Once the AUCAV enters the battle-space, threats such as dynamic time-sensitive targets may appear that the AUCAV and COCOM commander were not previously aware of. These dynamic time-sensitive targets will be discussed later when the problem instances are introduced. The goal of the AUCAV is to visit as many NAIs and targets as possible then return to the exit location within its allotted time limit.

Figure 1 displays an example of the benchmark scenario wherein the AUCAV knows all adversary arrival times and locations. All targets are deliberate time-sensitive targets. All SAM battery and NAI locations are known in advance of the mission. This scenario is labeled the benchmark because the AUCAV is aware of the arrival times and locations of all adversary targets and SAM batteries. Therefore, it provides the target selection route and cumulative payoff reward the AUCAV should expect for choosing the optimal route.

Though the benchmark scenario is solvable with extremely accurate intelligence information of adversary behavior, in reality, it can be hard to obtain such a high fidelity of information. It is difficult and computationally expensive to solve the optimal route without accurate knowledge of the adversary's behavior. Also, in SCAR missions, the AUCAV is not always aware of the arrival times and locations of all adversary targets and SAM batteries. Therefore, the AUCAV must obtain a target selection route close or ideally equivalent to the optimal route by executing a routing policy that anticipates the arrival times and locations of dynamic targets and SAM batteries in the battle-space.

3.1.3 Problem Instances

This thesis explores four problem instances that are excursions of the benchmark scenario where the AUCAV is not aware of the arrival times and locations of all adversary targets and SAM batteries. It must execute a routing policy that anticipates the arrival times and locations of dynamic targets and SAM batteries in hopes of discovering a target selection route that performs as close to the benchmark scenario's optimal route as possible. The first problem instance is the baseline instance wherein all of the SAM battery and NAI locations are known in advance of the mission but not all time-sensitive target locations are known. There are both a set of deliberate targets the AUCAV is aware of before the mission and a remaining set of dynamic time-sensitive targets arriving stochastically according to a Poisson distribution. The new targets that are anticipated to arrive into the battle-space are represented by a blue diamond as shown in Figure 2.

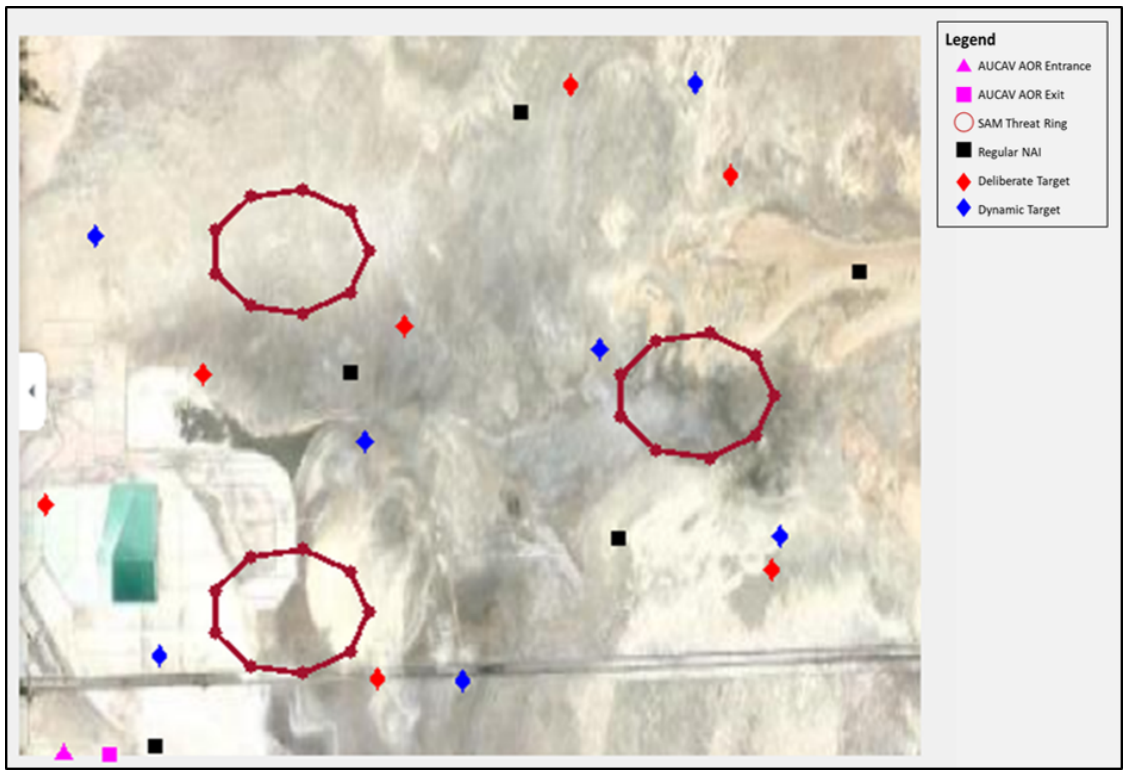

Figure 2: Baseline Instance

Figure 2 displays the starting state as known by the system and not the AUCAV. The blue diamonds represent the unknown dynamic targets that arrive one at a time through out the AUCAV's mission and remains for the duration of the mission. Once the target arrives it changes from an unknown dynamic target represented by the blue diamond to a known target represented by the red diamond. It remains a known target represented by the red diamond for the duration of the mission and is not removed from the battle-space unless the AUCAV reaches its location to destroy it prior to the end of the mission. This first instance addresses the question: How does the stochastic arrival of dynamic time-sensitive targets impact the AUCAV's mission performance?

The second instance introduces the arrival of SAM battery threat rings into the battle-space according to a Poison distribution. It addresses the question: How does

the stochastic arrival of dynamic time-sensitive targets and SAM battery threat rings impact the AUCAV's mission performance? This problem instance has the same parameters as the baseline instance but includes up to four additional SAM battery threat rings arriving into the battle-space during the AUCAV's mission.

The third problem instance is an extension of the baseline instance. All SAM battery threat rings are known prior to the mission and no additional ones arrive during the mission. Only time-sensitive targets arrive stochastically in a concentrated area. The question this instance addresses is how does the arrival of dynamic time-sensitive targets in a concentrated area impact the AUCAV's mission performance?

Finally, the last problem instance is a combination of the second and third instances where the SAM battery threat rings and time-sensitive targets arrive stochastically in a concentrated area. This instance addresses the question: How does the stochastic arrival of dynamic time-sensitive targets and SAM battery threat rings in a concentrated area impact the AUCAV's mission performance? Table 1 provides a summarized description of each problem instance.

Table 1: Summary Description of Instances

Instance 1	The AUCAV is aware of the location of deliberate TSTs, NAIs, and SAM batteries prior to the mission. During the mission, dynamic TSTs arrive on average every 10 minutes across the entire battle-space according to a Poison distribution.
Instance 2	Same as Instance 1 with one additional consideration: SAM batteries arrive on average every 35 minutes across the entire battle-space according to a Poison distribution during the mission.
Instance 3	Same as Instance 1 except the dynamic TSTs arrive in the northeast corner of the battle-space.
Instance 4	Same as Instance 2 except the dynamic TSTs and SAM batteries arrive in the northeast corner of the battle-space.

3.2 MDP Model

This section describes the problem formulation of the unmanned aircraft orienteering problem with stochastic obstacles and target arrivals as a discounted, infinite-horizon MDP model. The model consists of decision epochs, a state space, action sets, transitions, and rewards. The goal of this model is to determine an optimal route for the unmanned aircraft to travel that results in the highest cumulative reward. The highest cumulative reward can be achieved by destroying TSTs, visiting NAIs, and completing the mission by exiting the battle-space within an allotted time limit. This problem formulation takes into consideration the status and location of the entities (i.e., one AUCAV, dynamic and deliberate time-sensitive targets, SAM battery ring threats) at each decision epoch.

The decision epochs are represented by the point at which an event e happens that affects the status of the system prompting the AUCAV to make a decision.

Table 2: Event Types

e=1	Destruction of enemy TST (service)
e=2	AUCAV visits requested NAI (service)
e=3	Discovery of new dynamic TST (arrival)
e=4	Discovery of new SAM battery threat ring (arrival)
e=5	AUCAV destroyed by new SAM battery (destroyed)

Let $\mathcal{T} = \{0, 1, 2, ...\}$ denote the set of decision epochs at which decisions are made by the AUCAV. The events at which the AUCAV's decisions are prompted are described in Table 1.

3.2.1 The State Space

The state variable S_t describes the state of the system at decision epoch, t and is given by

$$S_t = (A_t, R_t, N_t, O_t, \tau, e) \in \mathcal{S} \tag{4}$$

wherein A_t represents the AUCAV's status, R_t represents the status of all known TSTs, N_t represents the status of all known NAIs, O_t represents the status of all SAM battery ring threats, τ is the current system time, e is the event type, and $\mathcal{S}$ is the set of all possible states.

The A_t variable represents the AUCAV status tuple. Let

$$A_t = (\ell_t^A, \rho_t), \tag{5}$$

wherein $\ell_t^A \in \mathbb{R}^2$ is the location of the AUCAV within a two dimensional battle space and ρ_t is the amount of playtime remaining for the AUCAV to exit the battle-space, successfully completing the mission. The AUCAV is assumed to be moving at a constant speed of 200 kilometers per hour.

The R_t variable represents the TST status tuple. Let

$$R_t = (\ell_{tr}^{R,TST}, \xi_{tr}, \zeta_r)_{r \in \mathcal{R}_t}, \tag{6}$$

wherein $\ell_{tr}^{R,TST} \in \mathbb{R}^2$ is the two dimensional location of target $r \in \mathcal{R}_t$ inwhich $\mathcal{R}_t$ is a set list of known targets. The $\xi_{tr} \in \{0, 1, \emptyset\}$ represents the TST status at time t. If $\xi_{tr} = 1$, the AUCAV is aware of the target and has not destroyed it by time t. If $\xi_{tr} = 0$, the AUCAV has destroyed the target. If $\xi_{tr} = \emptyset$, the target is not yet discovered by the AUCAV. When a TST is destroyed, it is removed from the set of known targets list, $\mathcal{R}_t$. The $\zeta \in \{0, 1\}$ represents the priority of the TSTs. If $\zeta_r = 0$, the TST r is a high payoff target (HPT). If $\zeta_r = 1$ the TST r is a high value target (HVT). An HVT is of higher value and priority than an HPT.

The N_t variable represents the NAI status tuple. Let

$$N_t = (\ell_n^{NAI}, \xi_{tn})_{n \in \mathcal{N}_t}, \tag{7}$$

wherein $\ell_n^{NAI} \in \mathbb{R}^2$ denotes the two dimensional location of NAI request $n \in \mathcal{N}_t$. It is assumed the locations of all NAI requests are known prior to the mission and no additional NAI requests will arrive through out the mission. However, $\xi_{tn} \in \{0, 1\}$ represents the NAI status at time t where 0 denotes the NAI was visited and 1 denotes AUCAV has not yet reached it. The AUCAV can choose to add or remove NAIs from its target route based on emerging priorities. Once an NAI is visited, it is removed from the set of requested NAIs, $\mathcal{N}_t$.

The O_t variable represents the obstacle transition node (OTN) (i.e., SAM battery threat ring) status tuple. Let

$$O_t = (\ell_{to_k}^{OTN})_{o \in \mathcal{O}_t, k \in \mathcal{K}} \tag{8}$$

where $\ell_{to_k}^{OTN} \in \mathbb{R}^2$ represents the two dimensional location of OTN $k \in \mathcal{K}$ on the circumference of SAM battery threat ring $o \in \mathcal{O}$ at time t. The size of $\mathcal{K}$ indicates the number of OTNs used to represent a SAM battery threat ring.

3.2.2 Action Space

The actions the AUCAV is allowed to take as a result of each event are listed in Table 2 below. The AUCAV is allowed to pursue all existing NAIs as well as deliberate and newly discovered dynamic TSTs. The OTN referenced in Table 2 is a node placed on the circumference of the SAM battery threat ring allowing the AUCAV to maneuver around the threat ring to reach its next assigned NAI or TST. The mathematical notation of the event types and action sets are addressed later in this chapter.

Table 3: Action Set

1	Update route to pursue newly selected TST, NAI, or OTN
2	Continue towards a previously selected TST, NAI, or OTN
3	Exit the AOR

The AUCAV may choose to pursue any TST, NAI, OTN, or the AOR exit location at any time t when t is less than the maximum playtime ($t < \rho_0$) and the AUCAV is not destroyed by a SAM battery ($e \neq 5$). The set of available actions is represented as

$$\mathcal{X}_{S_t} = (\ell_{tr}^{R,TST}, \xi_{tr})_{r \in R_t} \cup \ (\ell_n^{NAI}, \xi_{tn})_{n \in N_t} \cup \ (\ell_{to_k}^{OTN})_{o \in O_t, k \in K} \cup \ \{\Omega\}, \ \forall \ S_t \in \mathcal{S}, t \in \mathcal{T} \quad (9)$$

where Ω is the location of the AOR exit point the AUCAV must reach to end its mission. The AUCAV's selected destination $x_t \in \mathcal{X}_{S_t}$ at time t is considered feasible when $\xi_{tr} = 1$ and $\xi_{tn} = 1$ is true, indicating the TST request r is for a known TST not yet destroyed, and the NAI request n is for an NAI not yet visited by the AUCAV.

3.2.3 System Transition Function

The state transition for the AUCAV is represented by

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \quad (10)$$

where $S_t, S_{t+1} \in \mathcal{S}$ and $x_t \in \mathcal{X}_{S_t}$ holds true. Given the current state S_t, decision x_t, and considering the stochastic information W_{t+1} (i.e., SAM battery threat ring or TST arrival), the system model S^M evolves the system to a state representative of the AUCAV's chosen destination x_t, an enroute destination based on a new stochastic TST or OTN arrival, or a terminal state if the AUCAV is destroyed due to being within a SAM battery threat ring upon its arrival.

3.2.4 Contribution Function

The contribution function is represented by Equation 11 below for $S_t \in \mathcal{S}$. A positive reward φ^{NAI}, φ^{TST}, or φ^{Ω} is received for every NAI visited, TST destroyed, or when the AUCAV exits the AOR unscathed, respectively. A negative reward of $\varphi^{\Omega,OTR}$ is received if the AUCAV is inside a SAM battery threat ring (or obstacle threat ring (OTR)) upon its arrival. The AUCAV receives a reward of zero otherwise, including when it visits an OTN on the perimeter of a SAM battery threat ring.

$$C(S_t, x_t) = \begin{cases} \varphi^{TST,HVT}, & \text{if } x_t = (\ell_{tr}^{R,TST}, \xi_{tr}, \zeta_r),\ \ell_{tr}^{R,TST} = \ell_t^A,\ \xi_{tr} = 1,\ \zeta_t = 1 \\ \varphi^{TST,HPT}, & \text{if } x_t = (\ell_{tr}^{R,TST}, \xi_{tr}, \zeta_r),\ \ell_{tr}^{R,TST} = \ell_t^A,\ \xi_{tr} = 1,\ \zeta_t = 0 \\ \varphi^{NAI}, & \text{if } x_t = (\ell_{tn}^{NAI}, \xi_{tn}),\ \ell_{tn}^{NAI} = \ell_t^A,\ \xi_{tn} = 1 \\ \varphi^{\Omega,OTR}, & \text{if } \ell_t^A \in \text{polygon created by } (\ell_{to_k}^{OTN})_{k\in\mathcal{K}} \\ \varphi^{\Omega}, & \text{if } x_t = \{\Omega\},\ \Omega = \ell_t^A, \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $t > 0$, $t \in \mathcal{T}$, $x_t \in \mathcal{X}_{S_t}$, $\ell_t^A \in S_t$, $(\ell_{to_k}^{OTN})_{k\in\mathcal{K}} \in S_t$, and $S_t \in \mathcal{S}$ holds true.

3.2.5 Objective Function

The objective function is represented by Equation 12 where the policy that yields the highest expected total reward (ETR) (i.e. maximum $V(S_t)$) is recommend. The recommended policy π is a sequence of feasible locations (reference Equation 9) representing the optimal route the AUCAV should travel based on the state of the battle-space S_t at each decision epoch t that maximizes the AUCAV's ETR at the end of the mission.

$$V(S_t) = \max_{\pi\in\Pi} \mathbb{E}^{\pi}\Big[\sum_{t=1}^{\infty} C(S_t, X^{\pi}(S_t))\Big], \tag{12}$$

Equation 13 is the value function used to calculate the value of some fixed policy. This equation is used at each decision epoch t to evaluate the ETR the AUCAV is expected to receive at the end of the mission if it decides to travel to destination x_t at decision epoch t with only ρ_t playtime remaining for the AUCAV to complete the mission.

$$V(S_t) = C(S_t, x_t) + \mathbb{E}\big[V(S_{t+1})\big|S_t, x_t\big], \tag{13}$$

The Bellman equation, represented by Equation 14, is the mechanism used to attain an optimal policy. This equation uses Equation 13 to evaluate the ETR the AUCAV is expected to receive at the end of the mission if it decides to travel to destination x_t at decision epoch t. Every feasible destination x_t at decision epoch t is evaluated by Equation 13. Then Equation 14 is used to select the feasible destination x_t that results in the highest expected ETR at mission completion as the optimal policy based on the state of the battle-space S_t at decision epoch t.

$$V(S_t) = \max_{x_t \in \mathcal{X}_{S_t}} \Big\{C(S_t, x_t) + \mathbb{E}\big[V(S_{t+1})\big|S_t, x_t\big]\Big\}, \tag{14}$$

3.3 ADP Approach

In this thesis, an ADP approach is used to find a policy that on average produces a higher ETR than the benchmark policy. The benchmark policy is generated using the MDP model presented in Section 3.2. When an event occurs, the AUCAV selects the best decision given its knowledge of the current state of the environment. The benchmark policy does not consider what might happen in the future (i.e, dynamic target or SAM battery arrival) when selecting a decision in the present. Unlike the benchmark policy, the ADP approach enables the AUCAV to select decisions considering the future environment in the hopes of achieving a better ETR than the

benchmark policy. The MDP model's objective function (Equation 12) is used for the benchmark policy, and Equation 15 is the objective function for the ADP approach.

$$\max_{\theta} F(\theta) = \mathbb{E}^{\pi}\Big[\sum_{t=1}^{\infty} \hat{C}(S_t, X^{\pi}(S_t|\theta))\Big], \tag{15}$$

$$\max_{\theta} F(\theta) = \mathbb{E}^{\pi}\Big[\sum_{t=1}^{\infty} C(S_t, X^{\pi}(S_t|\theta)) + \sum_{f\in\mathcal{F}} \theta_f \phi_f(S_t, X^{\pi}(S_t|\theta))\Big], \tag{16}$$

Equation 16 exploits $\hat{C}(S_t, x_t) = C(S_t, x_t) + \Sigma_{f\in\mathcal{F}}\, \theta_f \phi_f(S_t, x_t)$ where the right hand side is referenced from the linear cost function correction Equation 2 introduced in Section 2.3. The benchmark policy only considers the immediate reward $C(S_t, x_t)$ while the ADP approach considers the immediate reward $C(S_t, x_t)$ and some additional cost function correction value $\Sigma_{f\in\mathcal{F}}\, \theta_f \phi_f(S_t, x_t)$ to calculate an ETR for some fixed policy x_t at decision epoch t. Equation 17 is the Bellman equation leveraged for the ADP approach.

$$\bar{F}(S_t|\theta) = \max_{x_t\in\mathcal{X}_{S_t}} \Big\{\hat{C}(S_t, x_t) + \mathbb{E}\big[\bar{F}(S_{t+1}|\theta)\big|S_t, X^{\pi}(S_t|\theta)\big]\Big\}, \tag{17}$$

The decision $x_t \in \mathcal{X}_{S_t}$ is determined using the decision function

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

As discussed in Section 2.3, the θ vector is a tunable parameter in which the ADP approach is used to find the θ-policy that yields on average the best ETR. If the best policy generated by the ADP approach does not display significant dependence on a CFA approach and $\theta = \bar{0}$, its ETR is approximately equivalent to the benchmark policy's ETR. Meaning the MDP approach objective function Equation 12 is equivalent to the ADP approach objective function Equation 15 when $\theta = \bar{0}$. The structure or

vector length of θ is determined by the structure of the basis function. Meaning the way we choose to structure ϕ as referenced in Equations 2 and 15 directly impacts the ETR and θ-policy recommended as a result of the ADP approach. The next section provides details on the structure of ϕ in the basic function.

3.3.1 Basic Function

The basic function uses coarse and tile coding to optimize the ETR the AUCAV receives at the end of its mission. The coarse coding involves the AOR being partitioned into multiple quadrants, and the tile coding is represented by tiles that reflect time intervals. For this thesis, the AOR is a $50 \times 50\text{km}^2$ battle-space, and the maximum playtime allowed for the AUCAV to finish its mission is 120 minutes. Tables 4 and 5 provide an example of when there are 4 time tiles and 16 quadrants given the $50 \times 50\text{km}^2$ battle-space and 120 minute time limit. It describes the characteristics belonging to each tile and quadrant as it relates to the AUCAV playtime ρ_t remaining when $t > 0$ and feasible destination coordinates $(x, y) \in \mathbb{R}^2$ in the battle-space, respectively.

Table 4: Tile Coding Characteristics

Tile 1	$0 \leq \rho_t \leq 30$
Tile 2	$30 < \rho_t \leq 60$
Tile 3	$60 < \rho_t \leq 90$
Tile 4	$90 < \rho_t \leq 120$

Table 5: Coarse Coding Characteristics

Quad 1	$0 \leq x \leq 12.5,$	$0 \leq y \leq 12.5$
Quad 2	$12.5 < x \leq 25,$	$0 \leq y \leq 12.5$
Quad 3	$25 < x \leq 37.5,$	$0 \leq y \leq 12.5$
Quad 4	$37.5 < x \leq 50,$	$0 \leq y \leq 12.5$
Quad 5	$0 \leq x \leq 12.5,$	$12.5 < y \leq 25$
Quad 6	$12.5 < x \leq 25,$	$12.5 < y \leq 25$
Quad 7	$25 < x \leq 37.5,$	$12.5 < y \leq 25$
Quad 8	$37.5 < x \leq 50,$	$12.5 < y \leq 25$
Quad 9	$0 \leq x \leq 12.5,$	$25 < y \leq 37.5$
Quad 10	$12.5 < x \leq 25,$	$25 < y \leq 37.5$
Quad 11	$25 < x \leq 37.5,$	$25 < y \leq 37.5$
Quad 12	$37.5 < x \leq 50,$	$25 < y \leq 37.5$
Quad 13	$0 \leq x \leq 12.5,$	$37.5 < y \leq 50$
Quad 14	$12.5 < x \leq 25,$	$37.5 < y \leq 50$
Quad 15	$25 < x \leq 37.5,$	$37.5 < y \leq 50$
Quad 16	$37.5 < x \leq 50,$	$37.5 < y \leq 50$

Equations 19-34 show a partial representation of the ϕ_f indicator functions where $f = (i, j)$ for $i \in \{1, 2, 3, 4..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$ denotes the ith quadrant of the jth tile. Decision $x_t \in \mathcal{X}_{S_t}$ is a feasible destination for the AUCAV to select at time t referenced from Equation 9 in Section 3.2.2. The remaining playtime is represented by $\rho_t \in S_t$ where $S_t \in \mathcal{S}$ holds true.

$$\phi_{1,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (0,0) \leq x_t \leq (12.5, 12.5) \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

$$\phi_{2,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (12.5,0) < x_t \leq (25, 12.5) \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

$$\phi_{3,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (25,0) < x_t \leq (37.5, 12.5) \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

$$\phi_{4,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (37.5,0) < x_t \leq (50, 12.5) \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

$$\phi_{5,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (0,12.5) \leq x_t \leq (12.5, 25) \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

$$\phi_{6,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (12.5,12.5) < x_t \leq (25, 25) \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

$$\phi_{7,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (25,12.5) < x_t \leq (37.5, 25) \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

$$\phi_{8,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (37.5,12.5) < x_t \leq (50, 25) \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

$$\phi_{9,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (0,25) \leq x_t \leq (12.5, 37.5) \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

$$\phi_{10,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (12.5, 25) < x_t \leq (25, 37.5) \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

$$\phi_{11,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (25, 25) < x_t \leq (37.5, 37.5) \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

$$\phi_{12,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (37.5, 25) < x_t \leq (50, 37.5) \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

$$\phi_{13,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (0, 37.5) x_t \leq (12.5, 50) \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

$$\phi_{14,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (12.5, 37.5) < x_t \leq (25, 50) \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

$$\phi_{15,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (25, 37.5) < x_t \leq (37.5, 50) \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

$$\phi_{16,1}(S_t, x_t) = \begin{cases} 1, & 0 \leq \rho_t \leq 30, (37.5, 37.5) < x_t \leq (50, 50) \\ 0, & \text{otherwise.} \end{cases} \tag{34}$$

Let Equations 19-34 also represent the same equations for Tiles 2, 3, and 4 in accordance with their constraints referenced in Table 4 for ρ_t. Figure 3 below shows a visual representation of the 16 quadrants. The coarse coding displayed by Figure 3 is used in each time tile.

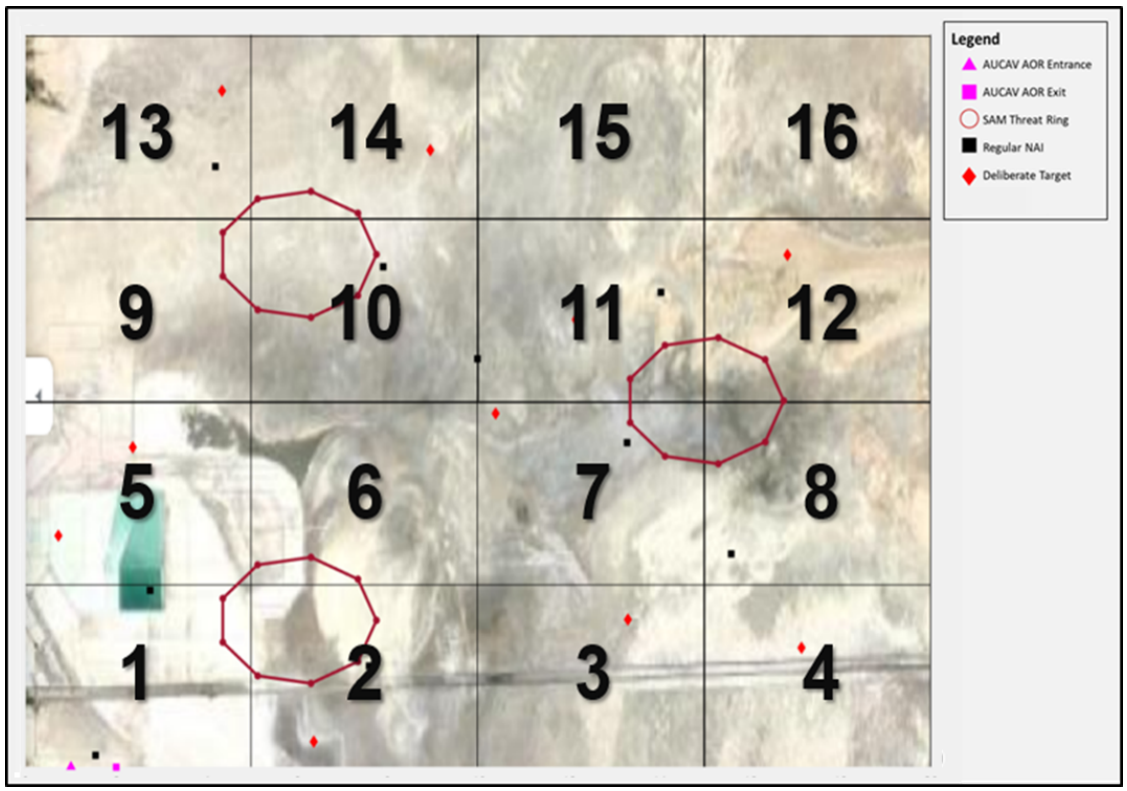

Figure 3: Coarse Coding Visual Example

Now that the structure of ϕ_f is established, two CFA approaches regarding the structure of θ_f and an additional adaptation to ϕ_f are explored in this thesis. The first approach combines the benchmark policy with a CFA using the ORTHOMADS algorithm represented by Equation 38 in which a clustering equation, the α parameter, and the variable vector θ^{MADS} are introduced.

The clustering equation involves summing all rewards of the feasible destinations $\mathcal{X}_{S_t}$ located in the same quadrant and time tile as x_t (a feasible destination that could be chosen by the AUCAV at decision epoch t). For example, let Figure 3 represent the state of the battle-space S_t at decision epoch t. Notice in Figure 3 there are two deliberate targets in quadrant 5. Imagine there is a third target in quadrant 5 and the feasible destination x_t being evaluated is the third target in quadrant 5, the target location x_t being evaluated is inputted into Equation 35.

$$\phi_{cluster}(S_t, x_t) = \begin{cases} \Sigma_{g \in G}\, C(S_t, x_t^g), & \text{if } x_t \neq (\ell_{to_k}^{OTN})_{k \in \mathcal{K}}, \\ & \rho_{\min}(j(\rho_t)) \leq \rho_t \leq \rho_{\max}(j(\rho_t)), \\ & x_t \geq \Big(x_{\min}(i(x_t)), y_{\min}(i(x_t))\Big), \\ & x_t \leq \Big(x_{\max}(i(x_t)), y_{\max}(i(x_t))\Big), \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

Let x_t^g refer to the gth feasible destination in the set of feasible destinations G located in the same quadrant and time tile as x_t where variables x_t and $(x_t^g)_{g \in G}$ meet the constraints shown in Equations 35. Let $y_{\min}(i(x_t))$, $y_{\max}(i(x_t))$, $x_{\min}(i(x_t))$, and $x_{\max}(i(x_t))$ denote the minimum and maximum latitude ($y \in \mathbb{R}$) and longitude ($x \in \mathbb{R}$) coordinate constraints associated with x_t being in the ith quadrant (reference Table 5). As a result of the example reference in Figure 3, if each of the targets in quadrant 5 are worth 100 points each, the following holds true where $G = \{1, 2\}$ represents the two feasible targets x_t^1 and x_t^2 in quadrant 5 and $\phi_{clustering}(S_t, x_t) = 200$ represents the additional expected reward the AUCAV can assume to receive for traveling to quadrant 5 after deciding to travel to target location x_t. Additionally, the constraint values in Equation 35 that reflect the boundaries of the battle-space region bordering quadrant 5 are $x_{\min}(i(x_t)) = 0$km longitude, $x_{\max}(i(x_t)) = 12.5$km longitude, $y_{\min}(i(x_t)) = 12.5$km latitude, and $y_{\max}(i(x_t)) = 25$km latitude in the 50×50km battle-space where $i = 5$.

The $\rho_{\min}(j(\rho_t))$ and $\rho_{\max}(j(\rho_t))$ represent the minimum and maximum playtime constraints associated with ρ_t being in the jth tile (reference Table 4). The decision $x_t \in \mathcal{X}_{S_t}$ is a feasible destination for the AUCAV to select at time t referenced from Equation 9 in Section 3.2.2. The $x_t \neq (\ell_{to_k}^{OTN})_{k \in \mathcal{K}}$ constraint is established to deter the

AUCAV from repetitively choosing OTN destinations falsely assuming an immediate reward of $C(S_t, x_t) + \Sigma_{f \in \mathcal{F}} \theta_f \phi_f(S_t, x_t)$ when the reward in reality is zero because $C(S_t, x_t) = 0$ when $x_t = \ell_{to_k}^{OTN}$ where the AUCAV travels to the kth OTN of the oth obstacle at decision epoch t. The exit destination (Ω) is still included to allow the algorithm to explore the possibility of a reward $\theta_{1j}^{\text{MADS}}$ when the AUCAV chooses to exit the battle-space or travel to feasible destinations near the exit destination in the jth time tile.

The α parameter is used as a scalar multiplier indicating how much the AUCAV takes into account the knowledge obtained from knowing the cumulative reward that could be obtained from traveling to a particular region (or quadrant) in a specific time frame. The following holds true where $\alpha \in \mathbb{R}$ for $\alpha > 0$. Thus α is any positive real number. The variable θ^{MADS} is used as an additional incentive reward to persuade the AUCAV to travel to a particular region of the battle-space when a certain amount of playtime is remaining. The clustering equation and the θ^{MADS} together are used to persuade the AUCAV to travel to the most valuable region of the battle-space at a specific amount of playtime remaining. Equation 35 shows the $\phi_{cluster}$ equation that must be added to the $\phi_{i,j}$ vector (reference Equations 19-34) in order to implement clustering with the coarse and tile coding equations. Equation 36 is the θ_f vector applied to this approach where α is a scalar multiplier of the $\phi_{cluster}$ value and the θ^{MADS} variable vector elements perform as scalar multipliers of the $\phi_{i,j}$ vector elements represented by Equations 19-34. The $\phi_{cluster}$ and $\phi_{i,j}$ are combined to create an updated ϕ_f as shown in Equation 37. Equation 38 combines the updated ϕ_f and θ_f (Equation 36) to demonstrate how the equations create the basis function used in the ADP approach that combines the benchmark policy with a CFA using the ORTHOMADS algorithm. The θ^{MADS} variable interacts with the ith quadrants and jth tiles indicator ϕ_{ij} functions while the α interacts with $\phi_{cluster}$ where

$\Sigma_{f\in\mathcal{F}}\,\theta_f\phi_f(S_t,x_t)$ referenced from Equation 16 continues to hold true as demonstrated by Equation 38.

$$\theta_f = [\,\alpha \quad \theta^{MADS}\,],\ \text{for } x_t \in \mathcal{X}_{S_t}, S_t \in \mathcal{S} \tag{36}$$

$$\phi_f(S_t,x_t) = \Big[\ \sum_{g\in G} C(S_t,x_t^g) \quad \phi_{ij}(S_t,x_t)\ \Big],\quad \text{for } x_t \in \mathcal{X}_{S_t}, S_t \in \mathcal{S}, \forall i,\ \ \forall j \tag{37}$$

$$\phi_f\theta_f = \begin{cases} [\alpha \sum_{g\in G} C(S_t,x_t^g)] + ... & \\ \sum_i \sum_j \theta_{ij}^{MADS}\phi_{ij}(S_t,x_t) & \text{if } x_t \neq (\ell_{to_k}^{OTN})_{k\in\mathcal{K}}, \\ & \rho_{\min}(j(\rho_t)) \leq \rho_t \leq \rho_{\max}(j(\rho_t)), \\ & x_t \geq \Big(x_{\min}(i(x_t)), y_{\min}(i(x_t))\Big), \\ & x_t \leq \Big(x_{\max}(i(x_t)), y_{\max}(i(x_t))\Big), \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

Here, $f \in cluster \times (i,j)$ where $i \in \{1,2,3,4...,14,15,16\}$ and $j \in \{1,2,3,4\}$ denotes the ith quadrant of the jth tile. Let the remaining variables represent the same values as described for Equation 35 above.

The second approach combines the benchmark policy with a CFA but does not use the ORTHOMADS algorithm with the θ^{MADS} vector. This approach is represented by Equation 39-41.

$$\theta_f = \alpha, \quad \text{for } \forall i,\ \ \forall j \tag{39}$$

$$\phi_f(S_t, x_t) = \sum_{g \in G} C(S_t, x_t^g), \quad \text{for } x_t \in \mathcal{X}_{S_t}, S_t \in \mathcal{S}, \forall i, \ \ \forall j \tag{40}$$

$$\phi_f \theta_f = \begin{cases} \Big[\alpha \sum_{g \in G} C(S_t, x_t)\Big], & \text{if } x_t \neq (\ell_{to_k}^{OTN})_{k \in \mathcal{K}}, \\ & \rho_{\min}(j(\rho_t)) \leq \rho_t \leq \rho_{\max}(j(\rho_t)), \\ & x_t \geq \Big(x_{\min}(i(x_t)), y_{\min}(i(x_t))\Big), \text{ and} \\ & x_t \leq \Big(x_{\max}(i(x_t)), y_{\max}(i(x_t))\Big), \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

Equations 39-40 are the same as Equations 36-37 except they do not include the additional reward incentive imposed by using the θ^{MADS} variable vector and ϕ_{ij} indicator functions vector with the ORTHOMADS algorithm. The second approach is created to delineate the difference in impact the θ^{MADS} and clustering function have on the ETR in comparison to the benchmark policy. This approach also allows for future research on sampling the initial θ and changing the value of α to see the impact it has on the ETR without completely depending on the ORTHOMADS algorithm. It also allows for the opportunity to create a novel heuristic ORTHOMADS-based algorithm in future studies.

3.3.2 Algorithmic Strategies

The algorithmic strategy associated with the benchmark policy is described by Algorithm 1 wherein a CFA approach is not used, and thus $\theta = \bar{0}$ is the policy being evaluated. There are N replications (or missions) of instances. Each instance's parameters, which describe dynamic TST or OTN arrival times and locations, are generated according to its associated random number generator string. The algorithm begins by allowing the AUCAV to review its environment (i.e., the locations of known

TSTs, NAIs, and OTNs) at epoch t for replication n and deciding which TST, NAI, or OTN destination x_t it should travel to next. Once the decision is made, the environment's evolution is simulated (see Figure 4) to the next epoch $t+1$. As a result, the estimated reward or contribution based on the AUCAV's decision is recorded at epoch $t+1$. A positive reward could be obtained if the AUCAV reached its destination prior to epoch $t+1$. A reward of zero is experienced if the AUCAV did not reach its destination or reached an OTN. A negative reward is received if a SAM battery arrives and the AUCAV is within its threat ring radius upon its arrival. If the AUCAV exits the battle-space by reaching the exit location or being destroyed by a SAM battery (i.e., $S_t = \Omega$), replication (or mission) n is complete, the applicable performance information is recorded for replication n, and replication $n+1$ begins.

Algorithm 1 Benchmark Algorithm

1: **for** $n = 1$ to N **do** (Policy Evaluation Loop) replications.
2: Initialize problem instance to begin trajectory if $n = 1$ or $S_{t+1,n-1} = \Omega$.
3: Generate a trajectory following next state $S_{t,n}$ (see Figure 4).
4: Determine decision x_t that optimizes $V(S_t)$ utilizing Equation 13.
5: Simulate transition to next pre-decision $S_{t+1,n}$.
6: Record contribution $C(S_{t+1,n}, x_t)$ using Equation 11.
7: If $S_{t+1,n} = \Omega$ record ETR estimate.
8: **end for**
9: Record mean ETR & 95% confidence interval (C.I.) half-width.
10: Record superlative ETR & Mission Completion Time.
11: Record overall algorithm run time.

The Benchmark-CFA with ORTHOMADS algorithm implements the same trajectory as Algorithm 1 but uses $\theta = [\ \alpha \quad \theta^{MADS}\ ]$ as the θ-policy CFA approach discussed above in Section 3.3.1. Algorithm 2 initializes an inputted α scalar value and θ^{MADS} vector value. The α scalar value may equal some fixed value or some value that changes based on the state of the system S_t but its value does not depend on and is not determined by the ORTHOMADS algorithm. The θ^{MADS} vector value, however, is dependent on the ORTHOMADS algorithm and updates each polling evaluation

up to V times where $V = \text{length}(\theta^{MADS}) + 10$. In some occasions, the polling evaluations may stop early and the algorithm will begin the next ORTHOMADS iteration (or policy improvement loop) before reaching V if the algorithm finds an improved solution while polling.

Algorithm 2 Benchmark-CFA with ORTHOMADS Algorithm

Initialize θ (linear model coefficients or weights).
for $m = 0$ to M **do** (Policy Improvement Loop) ORTHOMADS iterations.
    for v=1 to V **do** ORTHOMADS polling evaluations.
        for $n = 1$ to N **do** (Policy Evaluation Loop) replications.
            Initialize problem instance to begin trajectory if $n = 1$ or $S_{t,n-1} = \Omega$.
            Generate a trajectory following next state $S_{t,n}$ (see Figure 4).
            Determine decision x_t utilizing Equations 18 and 38.
            Simulate transition to next pre-decision $S_{t+1,n}$.
            Record contribution $C(S_{t+1,n}, x_{t,n})$ using Equation 11.
            If $S_{t+1,n} = \Omega$ record ETR estimate.
        end for
        Record optimal average ETR for ORTHOMADS polling evaluation v.
        Record optimal theta for ORTHOMADS polling evaluation v.
        Update θ utilizing ORTHOMADS orthogonal polling method.
    end for
    Record optimal average ETR for ORTHOMADS iteration m.
    Record optimal theta for ORTHOMADS iteration m.
    Update θ utilizing ORTHOMADS search method.
end for
Return the highest average ETR.
Return the θ that yields the highest average ETR.

In the occasion where the polling evaluations do not stop and cannot find an improved solution after the vth polling evaluation when $v = \text{length}(\theta^{MADS})$, 10 additional polling evaluations can occur. The 10 polling evaluations are used to ensure every single element of the θ^{MADS} vector is evaluated because the algorithm terminates when the maximum polling evaluations V are reached and there is no improved solution. Adding 10 additional polling evaluations prevents the algorithm from terminating if every element is evaluated but does not result in an improved solution and allows the opportunity for the algorithm to re-evaluate 10 of the θ^{MADS} vector elements with different values to search further for an improved solution before the

algorithm is terminated. For example, if some value h is added to each element of the θ^{MADS} vector per polling evaluation, once the last element is evaluated, some value $-h$ is then added to a previously evaluated element to explore if that new value will yield an improved solution. For more detailed information on these h values and their impact on the solution in each polling evaluation of the ORTHOMADS algorithm, please reference the work of Abramson et al. (2009).

During the development of this thesis, it was discovered that it is rare that each polling evaluation v produced a different mean TR unless that particular evaluation contained a θ^{MADS} vector value that produced a different and improved route than previous θ^{MADS} vector values in previous evaluations. The disadvantage of this is that an improved mean TR may be realized at a later polling evaluation. However, it is not worth the computational cost (i.e., additional run time) experienced for completing the entire polling evaluation, especially if an improved mean TR is not realized.

After the polling evaluation is complete and an improved solution is found, the ORTHOMADS algorithm conducts a search towards the next θ^{MADS} vector value utilizing what it has learned from the previous iteration m then begins the next iteration $m+1$. Due to computational effort experienced during each polling evaluation, the computational effort for each iteration is substantially high. Therefore, it is recommended to execute a low number of ORTHOMADS iterations and a reasonably high number of polling evaluations to address all of the applicable elements of the θ^{MADS} vector.

The Benchmark-CFA algorithm (without the ORTHOMADS algorithm) represented by Algorithm 3, only uses $\theta = \alpha$ as the θ-policy CFA approach discussed in Section 3.3.1 or offers the option of $\theta = [\ \alpha \quad \theta^{MADS}\ ]$ θ-policy using one θ^{MADS} vector value (preferably the best vector value $\theta^{*^{MADS}}$ outputted from Algorithm 2) as an additional incentive for the AUCAV to travel in certain regions of the battle-space.

Algorithm 3 Benchmark-CFA Algorithm

Initialize θ (linear model coefficients or weights).
for $n = 1$ to N **do** (Policy Evaluation Loop) replications.
    Initialize problem instance to begin trajectory if $n = 1$ or $S_{t,n-1} = \Omega$.
    Generate a trajectory following next state $S_{t,n}$ (see Figure 4).
    Determine decision x_t utilizing Equation 18 and 41.
    Simulate transition to next pre-decision $S_{t+1,n}$.
    Record contribution $C(S_{t+1,n}, x_{t,n})$ using Equation 11.
    If $S_{t+1,n} = \Omega$ record ETR.
end for
Record mean ETR & 95% C.I. half-width.
Record superlative ETR & Mission Completion Time.
Record overall algorithm run time.

Algorithm 3 is used to initialize the superlative $\theta^{*^{MADS}}$ vector value in conjunction with the clustering equation in which the θ^{MADS} policy never changes. When trying to obtain the sole impact the clustering equation CFA approach has on the ETR, the $\theta^{MADS} = \bar{0}$ indicating $\theta = \alpha$.

Table 6 summarizes a description of each algorithms' associated θ-policy structure, associated bellman equation structure utilized, and ADP approach further defined and discussed later in chapter 4.

Table 6: Summary Description of Algorithms

Algorithm	θ-policy	ADP Approach	Bellman Equation Used
Algorithm 1	$\theta = \bar{0}$	DROP policy	Equation 14
Algorithm 2	$\theta = [\alpha \quad \theta^{MADS}]$	DROP-CFA^{MADS}	Equation 17
Algorithm 3	$\theta = \alpha$	DROP-CFA	Equation 17
	$\theta = [\alpha \quad \theta^{*MADS}]$	DROP-CFA^{*MADS}	Equation 17

The deterministic, repeated, orienteering problem (DROP) policy is used for the benchmark policy Algorithm 1, the DROP-CFA^{MADS} approach is used for the Benchmark-CFA with ORTHOMADS Algorithm 2, and the DROP-CFA approach is used for the Benchmark-CFA without ORTHOMADS Algorithm 3. In this thesis, the DROP-CFA^{*MADS} approach is only used to evaluate the mission performance of the best

θ^{MADS} policies outputted by Algorithm 2. Once the superlative θ^{*MADS} vector values are determined it is inputted into Algorithm 3 to gain insight on how the AUCAV may perform in real time when using the superlative θ-policy in 10 missions (i.e. replications).

Algorithm 3 is used to compare the Benchmark-CFA and the Benchmark-CFA using ORTHOMADS' best θ^{*MADS} policy solution performance. It provides clarity on how much impact the clustering CFA equation and the θ^{MADS} vector value have on the ETR in comparison to the benchmark policy.

3.3.3 Simulation

A simulation is used to reflect the battle-space environment evolution when evolving between epoch t and epoch $t+1$ in accordance with the system transition function explained in Section 3.2.3. Figure 4 displays a visual graphic of this simulation (Goodwill, 2021).

Inputted into the simulation model are specific problem instance parameters, a decision rule represented by the actions the AUCAV chooses to take based on the θ-vector values along with their corresponding basis functions, and a random number stream. The key problem instance parameters are (1) the locations of the known TSTs, NAIs, and OTNs, (2) the arrival rate of unknown TSTs and OTNs, (3) the arrival locations of unknown TSTs and OTNs, and (4) the allotted playtime. The arrival rate and locations of unknown TSTs and OTNs are changed per each random number stream associated with each instance.

Once all parameters are inputted into the simulation, the AUCAV makes decisions in hopes of maximizing ETR as the system continues to evolve. The simulation is complete when the AUCAV is destroyed by a SAM battery or exits the battle-space successfully completing the mission. Then the ETR received and the θ_f associated

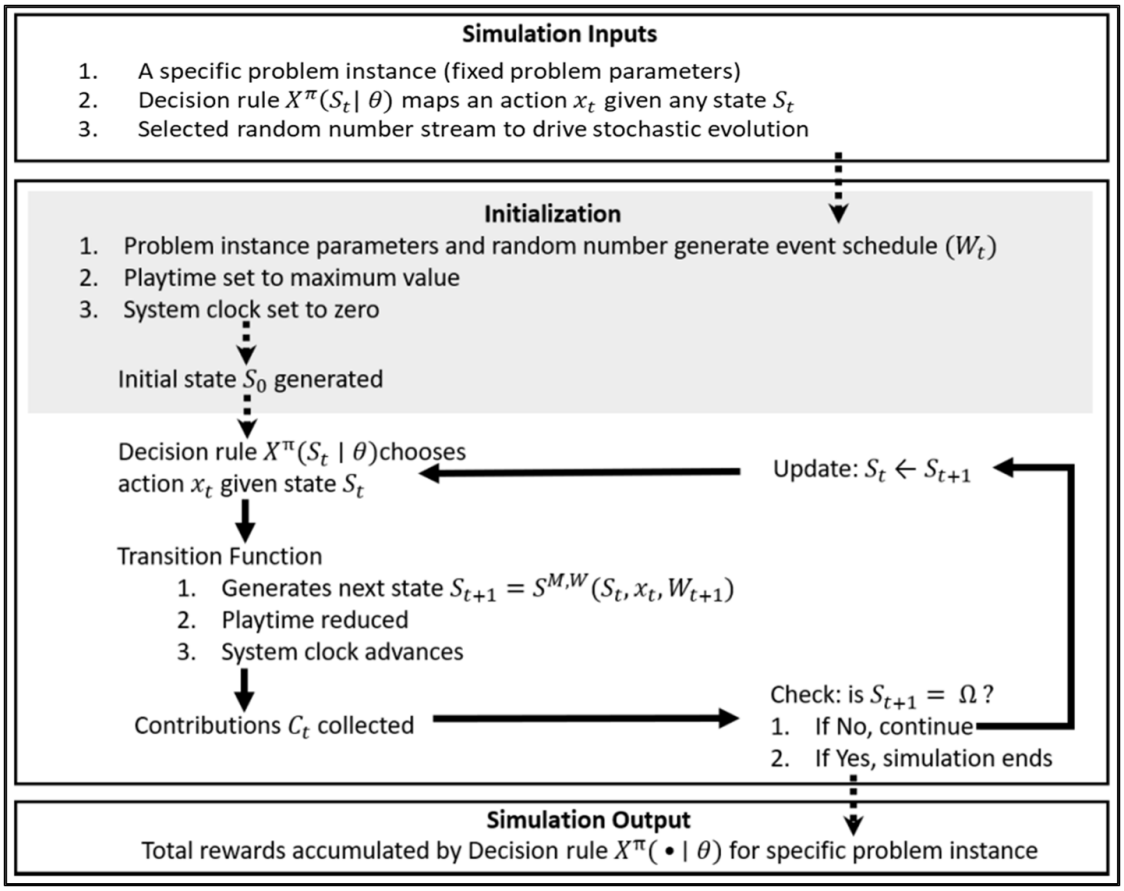

Figure 4: Simulation Model (from Goodwill (2021))

with that ETR are recorded.

3.3.4 Sampling and Exploration

Sampling and exploration is executed within the ORTHOMADS algorithm during its polling and searching process. The sampling takes place during the searching method. After the first iteration, each iteration of the ORTHOMADS algorithm thereafter searches for a random θ^{MADS} vector value. Within each iteration, the polling method is implemented where each element of the θ^{MADS} vector is changed by some value h. Thus, if the θ^{MADS} vector has 16 elements, there is at least 16 evaluations completed within one ORTHOMADS iteration where each element of θ^{MADS} is increased by the value of h every polling evaluation. The element that is increased

by h during its polling evaluation returns to its previous value when the next element is increased by h then evaluated. The orthogonality in the ORTHOMADS algorithm is addressed through the polling method when each element of θ^{MADS} is evaluated. The θ^{MADS} vectors that are evaluated via the polling method are orthogonal to the initial θ^{MADS} vector value at the beginning of each ORTHOMADS iteration.

Exploration is not considered in this thesis because it did not result in a significant benefit towards ETR results. However, for future improvement more sampling regarding discovering the best initial θ inputted into the algorithm should be studied.

3.3.5 Reward Engineering

The θ-modified objective functions used in the Benchmark-CFA approaches are created to incentivize certain decisions and behavior executed by the AUCAV. The benchmark policy objective function (Equation 12) does not include the CFA approach, thus the AUCAV's experienced ETR received by the end of its mission is equivalent to the benchmark policy's objective function's calculated ETR. However, when using the ADP approach Benchmark-CFA with or without the ORTHOMADS algorithm, the Benchmark-CFAs objective functions' calculated ETR is more likely to be of higher value than the AUCAV's experienced ETR received. This is due to adding the CFA cluster and θ^{MADS} variables to the benchmark policy's objective function to incentivize the AUCAV to travel to different destinations at different times than it would if it solely depended on the benchmark policy objective function.

For example, the θ_{ij}^{MADS} vector element associated with going to quadrant i in time tile j may equal a score of 100; the amount of points available at quadrant i in time tile j determined by the CFA cluster summation calculated by Equation 41 may equal 21; and the destination the AUCAV may consider going to is a TST worth 10 points located in quadrant i during time tile j. The benchmark policy's objective function

would recognize an immediate reward of 10 and the AUCAV would experience an immediate reward of 10 if it reaches its selected TST destination. The Benchmark-CFA objective function (without θ^{MADS} or $\theta^{MADS} = \bar{0}$) would recognize an immediate reward of 31 (10+21), and the AUCAV would only experience an immediate reward of 10 if it reaches its selected TST destination. The Benchmark-CFA objective function with $\theta^{MADS} = 100$ would recognize an immediate reward of 131 (10+21+100), and the AUCAV would only experience an immediate reward of 10 if it reaches its selected TST destination.

IV. Results and Analysis

4.1 Representative Scenario

The four instances discussed in Chapter 3 Section 3.1.3 each have assigned parameter settings. Table 7 lists the parameter settings explored in each instance.

Table 7: Problem Factors

Problem Factors to Study	Factor Levels
AUCAV Playtime, ρ_0	60 min , 120 min
Arrival Locations	Uniform, Northeast
SAM Arrival Rate per 35 mins, λ_{OTN}	1 SAM , 0 SAMs
Number of known SAMs	1 SAM , 2 SAMs
Fixed Factors	**Factor Level**
Target Arrival Rate per 10 mins, λ_{TST}	1
AOR Characteristics	50 km by 50 km
AUCAV Speed	108 Knots
Number of NAIs	5
Number of unknown SAMs	4
Number of Deliberate Targets	10
Number of Dynamic Target arrivals	20
Percentage of HVTs	50%
Reward Values	See Table 8

Goodwill (2021) provided Joint Forces Commander's (JFC's) approved reward values for TSTs and NAIs. Table 8 displays the rewards used in this thesis study in conjunction with some of JFC's approved rewards. The $\varphi^{\Omega,OTR}$ obstacle threat ring reward is a negative reward received when a SAM battery threat ring arrives and the AUCAV is within its threat ring radius upon its arrival. The $\varphi^{TST,HVT}$ and $\varphi^{TST,HPT}$ represent the rewards associated with the AUCAV destroying a high value target (HVT) and a high priority target (HPT), respectively. The φ^{NAI} reward is

associated with the AUCAV visiting an NAI destination, and the φ^{Ω} is associated with if the AUCAV exits the battle-space in conclusion of its mission.

Table 8: Rewards

$\varphi^{TST,HVT} = 100$
$\varphi^{TST,HPT} = 10$
$\varphi^{NAI} = 1$
$\varphi^{\Omega} = 0$
$\varphi^{\Omega,OTR} =$ -1000

The SAM threat ring arrival locations are locations near the adversary's deliberate and dynamic HVTs. This demonstrates that the adversary is knowledgeable enough to place their SAM batteries near their most valued assets. The SAM batteries will arrive around the deliberate HVTs first then the dynamic HVTs to illustrate that the adversary may have knowledge of the assets that the AUCAV could be aware of upon arrival into the battle-space. The longer the AUCAV is within the battle-space, the adversary believes the more likely the AUCAV will notice or become aware of their previously unknown high value assets. The scenarios and instances analyzed in this thesis ensures this idea is addressed.

Table 9: Algorithm Factors

Algorithm Factors to Study	Factor Levels
Number of policy improvement loops, M	1, 2, 3
Number of polling evaluation loops, V	42, 74, 138
Number of maximum tiles	4, 8
Number of maximum quadrants	8, 16
Scalar multiplier, α	5, ρ_t
Initial quadrant & tile reward policy, θ^{MADS}	See Equations 42-45
Fixed Factors	**Factor Level**
Number of replications, N	10

The algorithm parameter settings are displayed in Table 9. The initial quadrant & tile reward policy (θ^{MADS} vector) is created based on a strategy that forces the AUCAV to choose targets that are in the northern region of the battle-space when the remaining playtime is greater than one-fourth of the total allotted playtime. Once the remaining playtime is below one-fourth of the total allotted playtime, the AUCAV no longer depends on the θ^{MADS} incentive reward.

For a θ^{MADS} with a maximum of 4 time tiles and 8 quadrants, the initial quadrant and tile reward policy is represented by

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 5 \leq i \leq 8,\ \ 2 \leq j \leq 4 \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4\}$. For a θ^{MADS} with a maximum of 4 time tiles and 16 quadrants, the initial quadrant and tile reward policy is represented by

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16,\ \ 2 \leq j \leq 4 \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. For a θ^{MADS} with a maximum of 8 time tiles and 8 quadrants, quadrant and tile reward policy is represented by

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 5 \leq i \leq 8,\ \ 3 \leq j \leq 8 \\ 0, & \text{otherwise} \end{cases} \tag{44}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. Lastly, for a θ^{MADS} with a maximum of 8 time tiles and 16 quadrants, the initial quadrant and tile reward

policy is represented by

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16, \;\; 3 \leq j \leq 8 \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. The fixed factors in Tables 7 and 9 remain the same across all three algorithm approaches and all four problem instances. Some of the varying factors under study are implemented as fixed factors in the proposed ADP approaches and instances. This will be explained in detail when the experimental design structure is discussed later in this chapter.

4.2 Solution Methods

4.2.1 Benchmark

The benchmark solution method is the DROP, referenced from Gunawan et al. (2016), where a mixed integer linear program (MILP) model is used to solve the team orienteering problem (TOP). Goodwill (2021) used the same MILP model to solve a single entity orienteering problem resulting in the DROP policy solution method. Each decision $x_t \in \mathcal{X}_{S_t}$ is made based on an integer programming formulation as shown by Equations 46-52 below (Goodwill, 2021). The route to the first node in the optimal route proposed is chosen as the optimal decision x_t at epoch t. Where $x_t \in \mathcal{X}_{S_t}$ as referenced from Equation 9 is the next feasible destination (i.e. TST, NAI, OTN, or exit location) in the optimal route solved by the MILP model represented by Equations 46-52 below. Each time an event e happens that changes the state of the battle-space S_t this MILP model is solved to determine the new optimal route the AUCAV should travel based on the current battle-space environment at decision epoch t.

$$\textbf{Obj}: \max_{\boldsymbol{X_{ij}}} \quad \sum_{i=2}^{(|N|-1)} \sum_{j=2}^{|N|} P_j X_{ij}, \tag{46}$$

$$\text{s.t.} \quad \sum_{j=2}^{|N|} X_{1j} = \sum_{i=1}^{(|N|-1)} X_{i|N|} = 1, \tag{47}$$

$$\sum_{i=1}^{(|N|-1)} X_{ik} = \sum_{j=2}^{|N|} X_{kj} \leq 1; \ \text{ for } \ k = 2, ..., (|N|-1) \tag{48}$$

$$\sum_{i=1}^{(|N|-1)} \sum_{j=2}^{|N|} \rho_{ij} X_{ij} \leq \rho_t, \tag{49}$$

$$2 \leq u_i \leq |N|; \ \text{ for } \ i = 2, ..., |N|, \tag{50}$$

$$u_i - u_j + 1 \leq (|N|-1)(1 - X_{ij}); \ \text{ for } \ i = 2, ..., |N| \tag{51}$$

$$X_{ij} \in \{0, 1\}, \forall \ i, j \in N \tag{52}$$

The objective function's goal is to maximize ETR based on the current status of the system (Equation 46). Let $N = \{1, ..., |N|\}$ be the nodes represented as the set of feasible destination nodes $\mathcal{X}_{S_t}$ the AUCAV can travel to (i.e., the exit point and all known NAIs, OTNs, and TSTs in state S_t at epoch t). Where $N = 1$ is the source node that represents the AUCAV. P_j is the reward for visiting node j or feasible destination $x_t \in \mathcal{X}_{S_t}$. For the benchmark drop policy, the P_j rewards are referenced from the contribution function Equation 11 introduced in Section 3.2.4 and does not include the linear cost function correction (or basic functions) introduced in section 3.3.1. Equation 47 ensures the route starts at the AUCAV location and ends at the exit point. Equation 48 ensures each node is visited only once per route and the nodes of the route are sequential. Equation 49 ensures the time it takes to complete the route is less than the playtime remaining ρ_t. Equations 50 and 51 are subtour prevention constraints where $\{i+1, ..., (|N|-1)\} \equiv \{(\ell_{tn}^{NAI})_{n \in \mathcal{N}_t} \bigcup (\ell_{tr}^{R,HPT})_{r \in \mathcal{R}_t}\}$, $\bigcup \ \ell_{to_k}^{OTN}$ holds

true. Equation 52 enforces non-negativity.

4.2.2 Proposed Solution Methods

The first proposed solution method is the DROP policy with the CFA approach and ORTHOMADS algorithm (DROP-CFA^{MADS}) which uses the same MILP model formulation as the benchmark solution method except a linear cost function correction is incorporated into the objective function. Therefore, $P_j = C(S_t, j) + \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t, j)$ where the node j represents a feasible destination node $x_t \in \mathcal{X}_{S_t}$ in the route solved by the integer programming formulation is proposed. The formulation of this solution method can best be described by referencing Equations 16, 19-34, 36 and 38 discussed in Chapter 3.

The second proposed solution method is the DROP policy with the CFA approach not using ORTHOMADS (DROP-CFA), which uses the same formulation and similar objective function as the DROP-CFA^{MADS} approach. The only difference is θ^{MADS} is not included or is a fixed value due to not using the ORTHOMADS algorithm. The formulation of this solution method can best be described by referencing Equations 16, and 41 discussed in Chapter 3.

In both solution methods, a new formulation in which the basic function discussed in Section 3.3.1 is extended to include additional components that consider the playtime remaining, the playtime expended associated with a chosen feasible location, and the amount of surrounding TSTs associated with a chosen feasible location.

The DROP-CFA^{MADS} method is created to improve upon the DROP benchmark policy ETR solution but is potentially more computationally expensive. Therefore, the DROP-CFA method was created to explore a second approach with a reduced computational expense. This thesis uses the DROP-CFA^{MADS} method to determine the best policy $\theta^{*^{MADS}}$ and uses the DROP-CFA algorithm to gain more insight on

the clustering equation's effect (reference Equation 40 in Section 3.3.1) on mission performance.

4.3 Experimental Design

Tables 10-12 describe the experimental design used to determine the ETR performance differences between the DROP policy, DROP-CFA, and DROP-CFA^{MADS} approaches. The tables convey the 32 design runs for each of the two DROP-CFA approaches (i.e., 8 scenarios $\times$ 4 tile and quadrant combination runs). Therefore, 64 DROP-CFA approach runs and 8 DROP policy runs are implemented.

Table 10: Scenario Experimental Design
Problem Factors and Factor Levels

Scenario	Instance	Arrival Locations	# of known SAMs	$35\lambda_{SAM}$	ρ^{max}
1	1	Uniform	2	0	60
2	1	Uniform	2	0	120
3	2	Uniform	1	1	60
4	2	Uniform	1	1	120
5	3	Northeast	2	0	60
6	3	Northeast	2	0	120
7	4	Northeast	1	1	60
8	4	Northeast	1	1	120

The varying algorithm parameters for each DROP-CFA approach are described through Tables 11 and 12. Table 11 shows the DROP-CFA approach uses $\alpha = 5$. This was established through trial runs of Scenarios 1 and 3 with α =1, 3, 5, 7, and 10. These preliminary results showed that $\alpha = 5$ returned the highest mean TR.

In Table 12, the DROP-CFA^{MADS} approach uses $\alpha = \rho_t$. Trial runs of Scenarios 1 and 3 with $\alpha = \rho_t, \frac{1}{\rho_t}$, while setting θ^{MADS} in accordance with Equations 42-45, revealed that $\alpha = \rho_t$ performed significantly better.

Table 11: DROP-CFA Experimental Design per Scenario Algorithm Factors and Factor Levels

Run	Quadrants	Tiles	V	M	Initial θ^{MADS}	α
1	16	4	74	3	-	5
2	16	8	138	2	-	5
3	8	4	42	1	-	5
4	8	8	74	1	-	5

Table 12: DROP-CFA^{MADS} Experimental Design per Scenario Algorithm Factors and Factor Levels

Run	Quadrants	Tiles	V	M	Initial θ^{MADS}	α
1	16	4	74	3	Eq. 43	ρ_t
2	16	8	138	2	Eq. 45	ρ_t
3	8	4	42	1	Eq. 42	ρ_t
4	8	8	74	1	Eq. 44	ρ_t

4.4 Experimental Results

The DROP policy, DROP-CFA, and DROP-CFA^{MADS} approach executes 10 replications (or missions) per each design run. The mean TR for each design run is provided for each problem instance scenario. The run time for each design run is the total time it took the algorithm to complete 10 replications or missions. Therefore, the reported run time divided by 10 would be the average amount of time it took for the algorithm to complete one replication or mission. Thus, Algorithm 3, with a proposed CFA policy inputted into it, has the potential to perform computationally fast enough in real time to be used by an AUCAV in a real SCAR mission. The best TR and mission time experienced by the superlative replication for each design run is displayed to provide more insight.

4.4.1 DROP Benchmark Results

The DROP policy is used on all four instances, which are broken out into two scenarios each. The first and second scenario of each instance has a maximum playtime of 60 and 120 minutes, respectively. The fixed problem parameters for each scenario can be referenced in Table 7. The varying problem parameters displayed in Table 7 are assigned to each scenario in Table 10. Table 10 summarizes the changing parameters and problem features for each scenario. The mean TR, 95% confidence interval half-width, and algorithm run time for the DROP policy are displayed in Table 13.

Table 13: Benchmark Scenario Results

Scenario	Instance	Mean TR	Run Time
1	1	777±125.07	26.58 min
2	1	1043.9±118.2	45.28 min
3	2	737.9±135.68	23.06 min
4	2	785.2±468.3	88.13 min
5	3	902±132.41	32.41 min
6	3	1182.8±169.38	50.88 min
7	4	803.1±186.13	56.02 min
8	4	822.6±430.63	141.41 min

The trend worth highlighting is the significant difference between the first scenario and second scenario in Instances 1 and 3. Both scenarios include dynamic target arrivals with zero unknown SAM battery arrivals. Instance 1 has targets arriving uniformly across the entire AOR while Instance 3 has targets arriving in the northeast corner of the AOR. In both instances, the AUCAV's ability to destroy more targets and receive a higher ETR as its playtime increases is prevalent. Secondly, due to Instance 3 having targets arrive in a more concentrated area in the northeast corner of the AOR, its mean TR performance is slightly higher than Instance 1.

Instances 2 and 4 convey a slightly different trend. Both scenarios include dynamic target arrivals and unknown SAM battery arrivals. The unknown SAM battery ar-

rivals cause for a low difference between the mean TR performances of the first and second scenario of each instance. This result suggests an increase in the amount of playtime does not result in a significant increase in the mean TR due to the SAM battery's arrival near the high value targets. These general trends can be seen through out all of the ADP approach results as well.

The results show that additional replications per run and residual analysis could be done to explore these trends further while considering each scenario's best TR performance. Table 14 displays the best TR and simulated mission time associated with the best replication of each design run for the DROP Policy.

Table 14: Benchmark Superlative Results

Scenario	Instance	TR	Mission Time
1	1	1114	59.89 min
2	1	1275	119.52 min
3	2	1025	57.88 min
4	2	1284	119.2 min
5	3	1235	59.61 min
6	3	1515	119.69 min
7	4	1244	59.91 min
8	4	1515	119.9 min

Though additional replications are beneficial, 10 replications proved enough to begin to reveal the general trends amongst the instances and the ADP approaches discussed later in this chapter.

4.4.2 Problem Instance 1 - Target Arrivals

The proposed strategies in this section implement policies that imply an expected increase in high value target arrivals in emphasized quadrants and time tiles where if the AUCAV anticipates these arrivals then it will result in an improved TR.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with

Problem Instance 1, Scenario 1, is Design Run 4. It attains a best mean TR of 786.10 with a run time of 28.75 minutes. It is represented by Equation 53 with a maximum of 8 time tiles and 16 quadrants (reference Table 4.4.2).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16, \;\; 3 \leq j \leq 8 \\ 1000, & \text{if } i = 1, \;\; j = 5 \\ 0, & \text{otherwise} \end{cases} \tag{53}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. The initial θ^{MADS} policy strategy that forces the AUCAV to choose targets that are in the northern region of the battle-space when the remaining playtime is greater than one-fourth of the total allotted playtime remains the same here. However, the 1000 point reward for being in the 5th time tile and the 1st quadrant alludes to the inclusion of an additional strategy generating an improved mean TR. This strategy forces the AUCAV to choose targets in the southwest corner of the battle-space (reference Table 5) when the remaining playtime is within the time interval of greater than 30 to less than equal to 37.5 minutes. Thus forcing the AUCAV to proceed toward the exit location via the southwest corner of the battle-space in anticipation of new target arrivals.

Table 15: Problem Instance 1 DROP-CFA^{MADS} Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time[1]	Run Time[2]
1-1	8	4	60	737.40± 108.75	28.64 min	579.88 min
1-2	16	4	60	786.10± 125.57	28.84 min	501.30 min
1-3	8	8	60	734.70± 87.22	28.63 min	672.86 min
1-4	16	8	60	786.10± 125.57	28.75 min	493.74 min
2-1	8	4	120	1032.80± 122.91	45.52 min	1172.76 min
2-2	16	4	120	1033.60± 108.09	47.44 min	1059.66 min
2-3	8	8	120	1010.80± 130.20	50.33 min	1063.20 min
2-4	16	8	120	1034.90± 98.05	49.66 min	1145.30 min

[1] Run time for superlative θ^{MADS} policy to complete 10 replications.
[2] Run time for DROP-CFA^{MADS} policy search algorithm to evaluate 10 replications.

Table 4.4.2 shows the resulting mean TR of Design Run 4 is equivalent to Design Run 2. Given the ORTHOMADS algorithm process, it is worth noting that Design Run 4, with a θ^{MADS} vector size of 128 elements, has a shorter run time than Design Run 2, with a θ^{MADS} vector size of 64 elements (i.e., 4 tiles $\times$ 16 quadrants). The reason for this is because in Design Run 4 a locally optimal mean TR was found within a shorter amount of polling evaluation loops within each iteration than in Design Run 2. This is due to the algorithm starting the next ORTHOMADS iteration as soon as it finds a policy with a better mean TR than the previous ORTHOMADS iteration. However, if all of the polling evaluation loops were completed per ORTHOMADS iteration, Design Run 4 would have a higher run time than Design Run 2 since its total polling evaluation loops would be 138, and Design Run 2 would be 74. Reference Table 12 and the description of Algorithm 2 in Section 3.3.2 for more information on polling evaluation loops.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 1, Scenario 2, is Design Run 4. It attains a best mean TR of 1034.9 with a run time of 49.66 minutes. It is represented by Equation 54 with a maximum of 8 time tiles and 16 quadrants.

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16, \;\; 3 \leq j \leq 8 \\ 1000, & \text{if } i = 2, \;\; j = 3 \\ 3000, & \text{if } i = 2, \;\; j = 7 \\ 0, & \text{otherwise} \end{cases} \tag{54}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. Again, the initial θ^{MADS} policy strategy remains the same here. The 1000 point reward for being in the 3rd time tile and the 2nd quadrant alludes to an additional strategy. This strategy forces the AUCAV to choose targets in the region of the AOR that are below 12.5km latitude

and between 12.5 to 25km longitude in the $50\times50\text{km}^2$ battle-space (reference Table 5) when the remaining playtime is within the time interval of greater than 30 to less than equal to 45 minutes. The 3000 point reward for being in the 7th time tile and 2nd quadrant alludes to a second additional strategy that forces the AUCAV to choose targets in the same region of the AOR when the remaining playtime is within the time interval of greater than 90 to less than equal to 105 minutes. It is worth mentioning that the 3rd strategy associated with the 3000 point incentive reward indicates that it is three times more essential than the other strategies to execute because it implies that it results in a substantial increase in mission performance. All three strategies combined represent the θ^{MADS} superlative policy. Table 16 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA^{MADS} run in Instance 1.

Table 16: Problem Instance 1 DROP-CFA^{MADS} Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
1-1	8	4	60	1034	58.93 min
1-2	16	4	60	1123	59.92 min
1-3	8	8	60	935	58.29 min
1-4	16	8	60	1123	59.92 min
2-1	8	4	120	1375	115.99 min
2-2	16	4	120	1254	119.41 min
2-3	8	8	120	1265	118.28 min
2-4	16	8	120	1275	119.16 min

As shown in Table 17, the superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 1 Scenario 1 is Design Run 2. It attains a best mean TR of 767.30 with a run time of 26.48 minutes. It is represented by Equation 55 with a maximum of 4 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{55}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 4 time intervals when there is a maximum playtime of 60 minutes. Table 17 shows the resulting mean TR of Design Run 2 is equivalent to Design Run 4, indicating that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used in Scenario 1.

Table 17: Problem Instance 1 DROP-CFA Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time
1-1	8	4	60	728.70± 123.87	27.71 min
1-2	16	4	60	767.30± 104.06	26.48 min
1-3	8	8	60	728.70± 123.87	27.37 min
1-4	16	8	60	767.30± 104.06	26.67 min
2-1	8	4	120	1030.80± 111.74	45.52 min
2-2	16	4	120	1038.70± 127.80	45.99 min
2-3	8	8	120	1030.80± 111.74	45.72 min
2-4	16	8	120	1038.70± 127.80	47.08 min

The superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 1, Scenario 2, is Design Run 2. It attains a best mean TR of 1038.70 with a run time of 45.99 minutes. It is represented by Equation 56 with a maximum of 4 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{56}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 4 time intervals when there is a maximum playtime of 120 minutes. Table 17 shows the resulting mean TR of Design Run 2 is equivalent to Design Run 4, indicating that a policy that partitions the battle-space into 16 quadrants positively affects the mission

performance more than the number of time tiles used.

Table 18: Problem Instance 1 DROP-CFA Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
1-1	8	4	60	1034	58.97 min
1-2	16	4	60	1025	58.24 min
1-3	8	8	60	1034	58.97 min
1-4	16	8	60	1025	58.24 min
2-1	8	4	120	1265	119.99 min
2-2	16	4	120	1375	118.73 min
2-3	8	8	120	1265	119.99 min
2-4	16	8	120	1375	118.73 min

Table 18 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA run in Instance 1. Figure 5 displays the difference between the DROP policy and both DROP-CFA ADP approach performances for both scenarios of Instance 1.

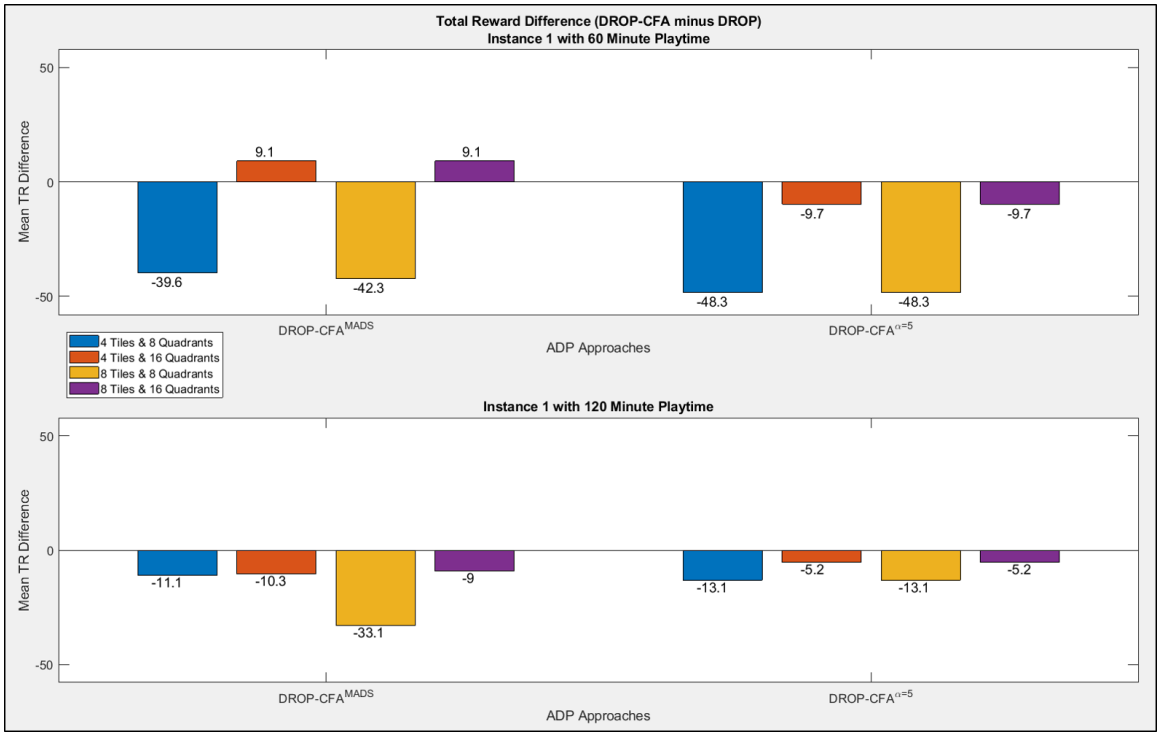

Figure 5: Instance 1 Results

A policy with 8 time tiles and 8 quadrants performs the worst, and a policy with 16 quadrants performs better than 8 quadrants. Both ADP approaches with 16 quadrants performs near the DROP policy performance or better. Only the DROP-CFA^{MADS} with 16 quadrants slightly performs better than the DROP policy.

4.4.3 Problem Instance 2 - Target & SAM Battery Arrivals

The proposed strategies in this section implement policies that imply an expected increase in high value target arrivals in emphasized quadrants and time tiles where if the AUCAV anticipates these arrivals then it will result in an improved TR. These strategies also imply that it is best to travel to the emphasized quadrants in a certain time tile to avoid the arrival of a SAM battery in a specific time tile or to travel to high value target locations in an emphasized quadrant before SAM batteries arrive that may blocked off the AUCAV from reaching the targets.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 2, Scenario 3, is Design Run 1. It attains a best mean TR of 734.50 with a run time of 24.95 minutes. It is represented by Equation 57 with a maximum of 4 time tiles and 8 quadrants (reference Table 4.4.3).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 5 \leq i \leq 8, \;\; 2 \leq j \leq 4 \\ 1000, & \text{if } i = 2, \;\; j = 3 \\ 0, & \text{otherwise} \end{cases} \tag{57}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4\}$. The initial θ^{MADS} policy strategy remains the same here. The 1000 point reward for being in the 3rd time tile and the 2nd quadrant alludes to an additional strategy that results in an improved mean TR. This strategy forces the AUCAV to choose targets in the region of the AOR that are below 25km latitude and between 12.5 to 25km longitude in the $50 \times 50 \text{km}^2$ battle-space (reference Table 5) when the remaining playtime is within the time interval of greater than 30 to less than equal to 45 minutes.

This strategy implies that there is an expected increase in high value target arrivals in quadrant 2 when there is 30-45 minutes of playtime remaining for the AUCAV to complete the mission. If the AUCAV anticipates these arrivals then it will result in an improved TR. This also implies that it is best to travel to quadrant 2 before there is less than 30 minutes of playtime remaining to complete the mission because a SAM battery may arrive later in the mission causing the AUCAV to miss the opportunity to travel to high value target locations in quadrant 2 that are not blocked off by SAM battery threat rings.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 2, Scenario 4, is Design Run 2. It attains a best mean TR of 955.8 with a run time of 108.37 minutes. It is represented by Equation 58 with a maximum

Table 19: Problem Instance 2 DROP-CFA^{MADS} Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time[1]	Run Time[2]
3-1	8	4	60	734.50± 128.74	24.95 min	123.82 min
3-2	16	4	60	730.20± 114.38	23.83 min	99.29 min
3-3	8	8	60	719.50± 95.66	23.84 min	146.07 min
3-4	16	8	60	730.20± 114.38	23.30 min	102.49 min
4-1	8	4	120	580.10± 554.97	99.00 min	214.46 min
4-2	16	4	120	955.80± 123.04	108.37 min	207.25 min
4-3	8	8	120	745.50± 436.28	99.25 min	209.36 min
4-4	16	8	120	944.90± 153.45	108.09 min	251.36 min

[1] Run time for superlative θ^{MADS} policy to complete 10 replications.
[2] Run time for DROP-CFA^{MADS} policy search algorithm to evaluate 10 replications.

of 4 time tiles and 16 quadrants (reference Table 4.4.3).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16,\ \ 3 \leq j \leq 4 \\ 1000, & \text{if } i = 1,\ \ j = 2 \\ 9000, & \text{if } i = 2,\ \ j = 2 \\ 3000, & \text{if } i = 5,\ \ j = 3 \\ 0, & \text{otherwise} \end{cases} \tag{58}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. The initial θ^{MADS} policy strategy remains the same here. The 1000 point reward for being in the 2nd time tile and the 1st quadrant alludes to a strategy that forces the AUCAV to choose targets in the region of the AOR that are below 12.5km latitude and 12.5km longitude in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 30 to less than equal to 60 minutes. The 9000 point reward for being in the 2nd time tile and the 2nd quadrant alludes to a strategy that forces the AUCAV to choose targets in the region of the AOR that are below 12.5km latitude and between 12.5 to 25km longitude in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 30 to less than equal to 60 minutes. The

3000 point reward for being in the 3rd time tile and the 5th quadrant alludes to a strategy that forces the AUCAV to choose targets in the region of the AOR that are between 12.5 to 25km latitude and below 12.5km longitude in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 60 to less than equal to 90 minutes. All four strategies combined represent the superlative θ^{MADS} policy. Table 20 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA^{MADS} run in Instance 2.

Table 20: Problem Instance 2 DROP-CFA^{MADS} Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
3-1	8	4	60	1025	59.92 min
3-2	16	4	60	1025	59.94 min
3-3	8	8	60	935	58.80 min
3-4	16	8	60	1025	59.94 min
4-1	8	4	120	1184	118.45 min
4-2	16	4	120	1144	119.57 min
4-3	8	8	120	1145	119.67 min
4-4	16	8	120	1175	116.53 min

Table 21 shows the superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 2, Scenario 3, is Design Run 4. It attains a best mean TR of 749.30 with a run time of 22.64 minutes. It is represented by Equation 59 with a maximum of 8 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{59}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. This suggests that it is best to partition the 50×50km^2 battle-space into 16 regions and split the playtime into 8 time intervals when there is a maximum playtime of 60 minutes. Table 21 shows the resulting mean TR of Design Run 4 is equivalent to Design Run 2, indicating

that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used.

Table 21: Problem Instance 2 DROP-CFA Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time
3-1	8	4	60	723.60± 129.76	22.05 min
3-2	16	4	60	749.30± 116.63	22.83 min
3-3	8	8	60	723.60± 129.76	22.08 min
3-4	16	8	60	749.30± 116.63	22.64 min
4-1	8	4	120	464.70± 615.52	58.56 min
4-2	16	4	120	650.00± 508.26	91.38 min
4-3	8	8	120	464.70± 615.52	58.84 min
4-4	16	8	120	650.00± 508.26	91.12 min

The superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 2, Scenario 4, is Design Run 4. It attains a best mean TR of 650 with a run time of 91.12 minutes. It is represented by Equation 60 with a maximum of 8 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{60}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 8 time intervals when there is a maximum playtime of 120 minutes. Table 21 shows the resulting mean TR of Design Run 4 is equivalent to Design Run 2. Again, indicating that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used. Table 22 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA design run in Instance 2.

Table 22: Problem Instance 2 DROP-CFA Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
3-1	8	4	60	945	59.20 min
3-2	16	4	60	1025	59.70 min
3-3	8	8	60	945	59.20 min
3-4	16	8	60	1025	59.70 min
4-1	8	4	120	1165	119.00 min
4-2	16	4	120	1144	119.53 min
4-3	8	8	120	1165	119.00 min
4-4	16	8	120	1144	119.53 min

Figure 6 displays the difference between the DROP policy and both DROP-CFA ADP approaches' performances for both scenarios of Instance 2.

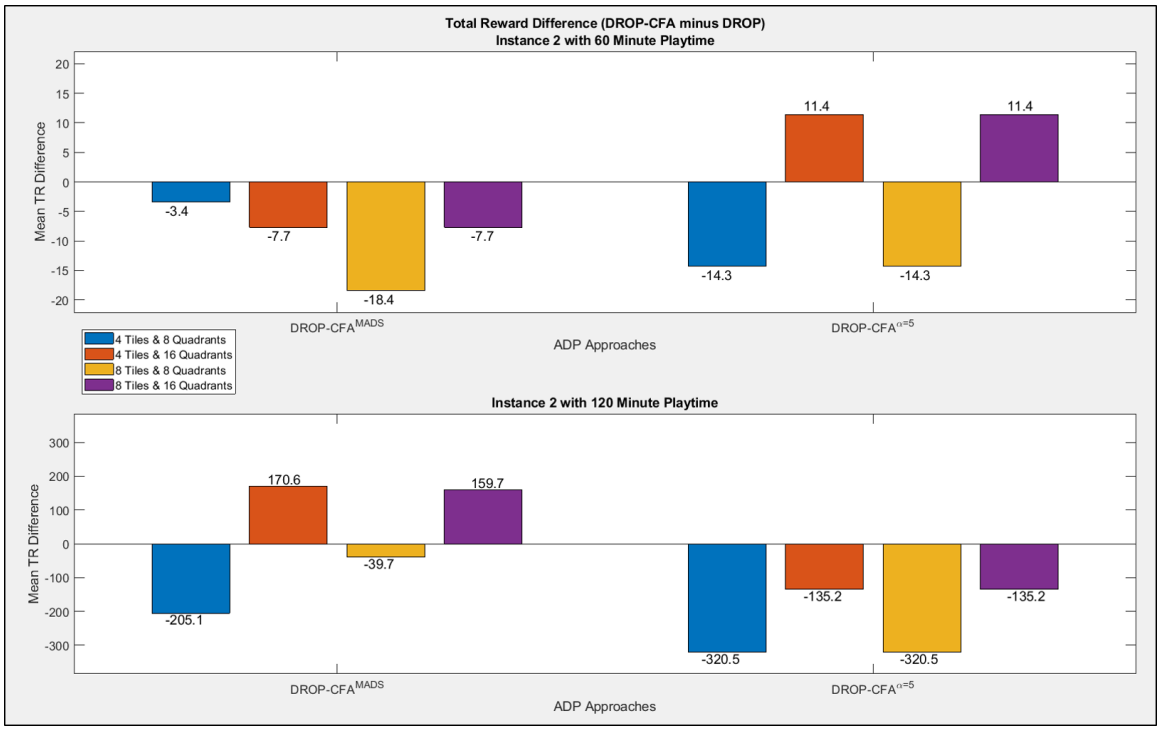

Figure 6: Instance 2 Results

For Instance 2 Scenario 3 the DROP-CFA approach with 16 quadrants performs slightly better than the DROP policy and DROP-CFA^{MADS} approach. For Instance 2 Scenario 4 the DROP-CFA^{MADS} approach with 16 quadrants performs better than the DROP policy and DROP-CFA approach.

4.4.4 Problem Instance 3 - Target Arrivals in Concentrated Area

Similar to Instance 1, the proposed strategies in this section implement policies that imply an expected increase in high value target arrivals in emphasized quadrants and time tiles where if the AUCAV anticipates these arrivals then it will result in an improved TR.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 3, Scenario 5, is Design Run 2. It attains a best mean TR of 898.20 with a run time of 34.66 minutes. It is represented by Equation 61 with a maximum of 4 time tiles and 16 quadrants (reference Table 4.4.4).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16, \;\; 2 \leq j \leq 4 \\ 0, & \text{otherwise} \end{cases} \tag{61}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. The initial θ^{MADS} policy strategy remains the same here with no other additional strategies added. This initial policy strategy forces the AUCAV to choose targets in the northern region of the AOR that are above 25km latitude in the $50 \times 50 km^2$ battle-space when the remaining playtime is within the time interval of greater than 15 to less than equal to 60 minutes. This superlative θ^{MADS} policy reveals that the algorithm could not find a better policy than the initial θ^{MADS} policy.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 3, Scenario 6, is Design Run 1. It attains a best mean TR of 1182.90

Table 23: Problem Instance 3 DROP-CFA^{MADS} Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time[1]	Run Time[2]
5-1	8	4	60	859.60± 119.88	36.67 min	537.56 min
5-2	16	4	60	898.20± 132.61	34.66 min	553.09 min
5-3	8	8	60	859.60± 119.88	35.35 min	619.69 min
5-4	16	8	60	898.20± 132.61	35.34 min	563.95 min
6-1	8	4	120	1182.90± 169.55	53.99 min	869.60 min
6-2	16	4	120	1160.70± 182.24	53.37 min	869.60 min
6-3	8	8	120	1169.00± 163.27	53.49 min	1004.19 min
6-4	16	8	120	1160.70± 182.24	54.43 min	1004.19 min

[1] Run time for superlative θ^{MADS} policy to complete 10 replications.
[2] Run time for DROP-CFA^{MADS} policy search algorithm to evaluate 10 replications.

with a run time of 53.99 minutes. It is represented by Equation 62 with a maximum of 4 time tiles and 8 quadrants (reference Table 4.4.4).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 5 \leq i \leq 8, \;\; 2 \leq j \leq 4 \\ 3000, & \text{if } i = 3, \;\; j = 2 \\ 1000, & \text{if } i = 3, \;\; j = 4 \\ 0, & \text{otherwise} \end{cases} \tag{62}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4\}$. The initial θ^{MADS} policy strategy remains the same here. The 3000 point reward for being in the 2nd time tile and the 3rd quadrant alludes to a strategy that forces the AUCAV to choose targets in the region of the AOR that are below 25km latitude and between 25 to 37.5km longitude in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 30 to less than equal to 60 minutes. The 1000 point reward for being in the 4th time tile and the 3rd quadrant alludes to a strategy that forces the AUCAV to choose targets in the same region previously mentioned in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 90 to less than equal to 120 minutes. All three strategies combined represent the superlative

θ^{MADS} policy for Scenario 6. Table 24 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA^{MADS} run in Instance 3.

Table 24: Problem Instance 3 DROP-CFA^{MADS} Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
5-1	8	4	60	1235	59.57 min
5-2	16	4	60	1235	59.57 min
5-3	8	8	60	1235	59.57 min
5-4	16	8	60	1235	59.57 min
6-1	8	4	120	1235	59.57 min
6-2	16	4	120	1235	59.57 min
6-3	8	8	120	1235	59.57 min
6-4	16	8	120	1235	59.57 min

Table 25 shows the superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 3, Scenario 5, is Design Run 2. It attains a best mean TR of 885.60 with a run time of 31.43 minutes. It is represented by Equation 63 with a maximum of 4 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{63}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 4 time intervals when there is a maximum playtime of 60 minutes. Table 25 shows the resulting mean TR of Design Run 2 is equivalent to Design Run 4. Indicating, again, that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used.

The superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 3, Scenario 6, is Design Run 2. It attains a best mean TR of 1181.90 with a run time of 50.88 minutes. It is represented by Equation 64 with a maximum of 4

Table 25: Problem Instance 3 DROP-CFA Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time
5-1	8	4	60	858.70± 130.19	31.55 min
5-2	16	4	60	885.60± 138.46	31.43 min
5-3	8	8	60	858.70± 130.19	32.64 min
5-4	16	8	60	885.60± 138.46	32.50 min
6-1	8	4	120	1145.90± 169.52	51.12 min
6-2	16	4	120	1181.90± 169.02	50.88 min
6-3	8	8	120	1145.90± 169.52	51.94 min
6-4	16	8	120	1181.90± 169.02	51.92 min

time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{64}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 4 time intervals when there is a maximum playtime of 120 minutes. Table 25 shows the resulting mean TR of Design Run 2 is equivalent to Design Run 4. Again, indicating that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used. Table 26 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA design run in Instance 3.

Figure 7 displays the difference between the DROP policy and both DROP-CFA ADP approaches' performances for both scenarios of Instance 3.

Table 26: Problem Instance 3 DROP-CFA Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
5-1	8	4	60	1235	59.29 min
5-2	16	4	60	1244	59.53 min
5-3	8	8	60	1235	59.29 min
5-4	16	8	60	1244	59.53 min
6-1	8	4	120	1515	119.68 min
6-2	16	4	120	1515	117.69 min
6-3	8	8	120	1515	119.68 min
6-4	16	8	120	1515	117.69 min

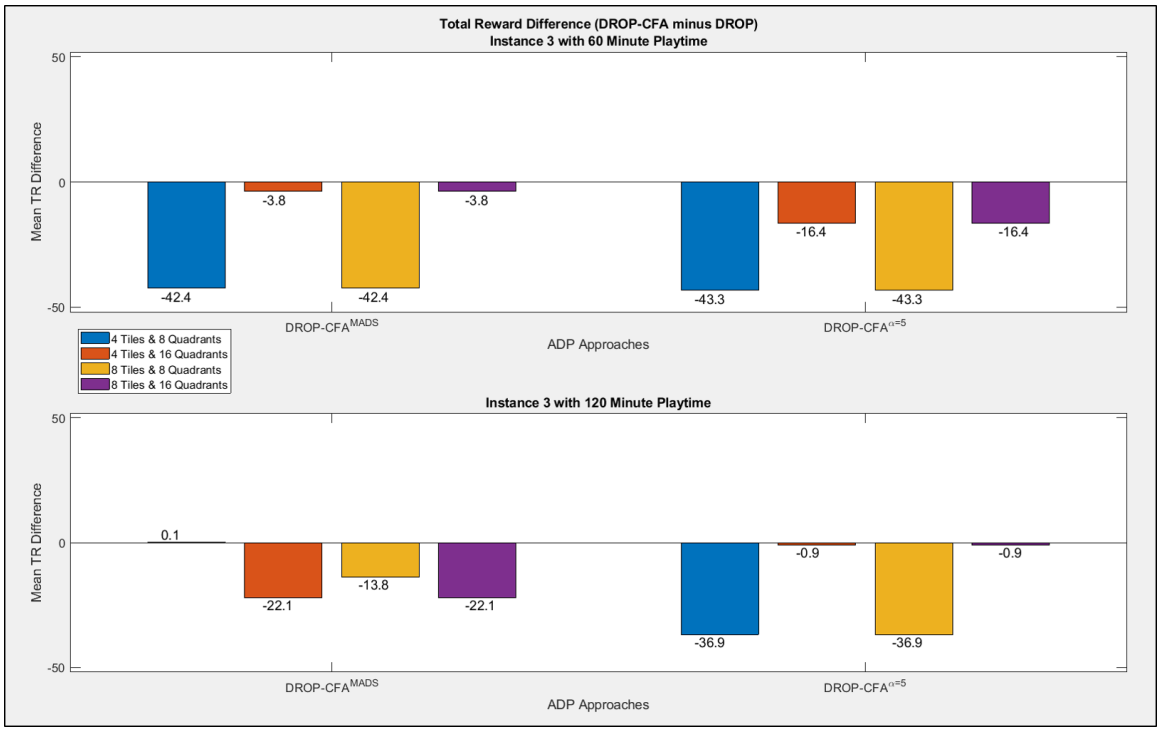

Figure 7: Instance 3 Results

For Instance 3, the benchmark DROP policy performs nearly equivalent to or better than both ADP approaches.

4.4.5 Problem Instance 4 - Target & SAM Battery Arrivals in Concentrated Area

Similar to instance 2, the proposed strategies in this section implement policies that imply an expected increase in high value target arrivals in emphasized quadrants and time tiles where if the AUCAV anticipates these arrivals then it will result in an improved TR. These strategies also imply that it is best to travel to the emphasized quadrants in a certain time tile to avoid the arrival of a SAM battery in a specific time tile or to travel to high value target locations in an emphasized quadrant before SAM batteries arrive that may blocked off the AUCAV from reaching the targets.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with Problem Instance 4, Scenario 7, is Design Run 2. It attains a best mean TR of 813.10 with a run time of 53.17 minutes. It is represented by Equation 65 with a maximum of 4 time tiles and 16 quadrants (reference Table 4.4.5).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 9 \leq i \leq 16, \;\; 2 \leq j \leq 4 \\ 1000, & \text{if } i = 2, \;\; j = 3 \\ 0, & \text{otherwise} \end{cases} \tag{65}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. The initial θ^{MADS} policy strategy remains the same. The 1000 point reward for being in the 3rd time tile and the 2nd quadrant alludes to a strategy that forces the AUCAV to choose targets that are in the region below 12.5km latitude and between 12.5 to 25km longitude in the $50 \times 50 \text{km}^2$ battle-space when the remaining playtime is within the time interval of greater than 30 to less than equal to 45 minutes. Both strategies combined represent the superlative θ^{MADS} policy for Scenario 7.

The superlative policy θ^{MADS} for the DROP-CFA^{MADS} approach associated with

Table 27: Problem Instance 4 DROP-CFA^{MADS} Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time[1]	Run Time[2]
7-1	8	4	60	762.50± 196.07	61.43 min	1004.19 min
7-2	16	4	60	813.10± 188.37	53.17 min	1394.41 min
7-3	8	8	60	749.70± 150.04	55.98 min	1394.41 min
7-4	16	8	60	803.00± 189.79	52.57 min	1394.41 min
8-1	8	4	120	731.10± 545.84	148.65 min	1394.41 min
8-2	16	4	120	828.10± 454.05	155.88 min	1394.41 min
8-3	8	8	120	863.10± 488.00	155.93 min	1394.41 min
8-4	16	8	120	828.10± 454.05	155.44 min	1394.41 min

[1] Run time for superlative θ^{MADS} policy to complete 10 replications.
[2] Run time for DROP-CFA^{MADS} policy search algorithm to evaluate 10 replications.

Problem Instance 4, Scenario 8, is Design Run 3. It attains a best mean TR of 863.10 with a run time of 155.93 minutes. It is represented by Equation 66 with a maximum of 8 time tiles and 8 quadrants (reference Table 4.4.5).

$$\theta_{i,j}^{MADS} = \begin{cases} 1000, & \text{if } 5 \leq i \leq 8, \;\; 3 \leq j \leq 8 \\ 1000, & \text{if } i = 4, \;\; j = 2 \\ 0, & \text{otherwise} \end{cases} \tag{66}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. The initial θ^{MADS} policy strategy remains the same. The 1000 point reward for being in the 2nd time tile and the 4th quadrant alludes to a strategy that forces the AUCAV to choose targets that are in the region below 25km latitude and greater than 37.5km longitude in the 50×50km^2 battle-space when the remaining playtime is within the time interval of greater than 15 to less than equal to 30 minutes. Both strategies combined represent the superlative θ^{MADS} policy for Scenario 8. Table 28 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA^{MADS} run in Instance 4.

Table 29 shows the superlative policy θ^{α} for the DROP-CFA approach associated

Table 28: Problem Instance 4 DROP-CFA^{MADS} Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
7-1	8	4	60	1174	59.84 min
7-2	16	4	60	1174	59.84 min
7-3	8	8	60	1174	59.84 min
7-4	16	8	60	1174	59.84 min
8-1	8	4	120	1174	59.84 min
8-2	16	4	120	1174	59.84 min
8-3	8	8	120	1174	59.84 min
8-4	16	8	120	1174	59.84 min

with Problem Instance 4, Scenario 7, is Design Run 2. It attains a best mean TR of 811.10 with a run time of 56.42 minutes. It is represented by Equation 67 with a maximum of 4 time tiles and 16 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{67}$$

where $i \in \{1, 2, 3, ..., 14, 15, 16\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to partition the $50 \times 50\text{km}^2$ battle-space into 16 regions and split the playtime into 4 time intervals when there is a maximum playtime of 60 minutes. Table 29 shows the resulting mean TR of Design Run 2 is equivalent to Design Run 4. Indicating, again, that a policy that partitions the battle-space into 16 quadrants positively affects the mission performance more than the number of time tiles used.

The superlative policy θ^{α} for the DROP-CFA approach associated with Problem Instance 4, Scenario 8, is Design Run 1. It attains a best mean TR of 967.70 with a run time of 187.27 minutes. It is represented by Equation 68 with a maximum of 4 time tiles and 8 quadrants.

$$\theta^{\alpha}_{i,j} = 5, \quad \text{for } \forall i, \ \forall j \tag{68}$$

where $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $j \in \{1, 2, 3, 4\}$. This suggests that it is best to

Table 29: Problem Instance 4 DROP-CFA Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	Mean TR	Run Time
7-1	8	4	60	766.80± 158.33	52.98 min
7-2	16	4	60	811.10± 177.17	56.42 min
7-3	8	8	60	766.80± 158.33	53.80 min
7-4	16	8	60	811.10± 177.17	56.56 min
8-1	8	4	120	967.70± 265.29	187.27 min
8-2	16	4	120	868.00± 516.40	158.16 min
8-3	8	8	120	967.70± 265.29	187.63 min
8-4	16	8	120	868.00± 516.40	159.30 min

partition the $50 \times 50\text{km}^2$ battle-space into 8 regions and split the playtime into 4 time intervals when there is a maximum playtime of 120 minutes. Table 29 shows the resulting mean TR of Design Run 1 is equivalent to Design Run 2, indicating that a policy that partitions the battle-space into 8 quadrants positively affects the mission performance more than the number of time tiles used. Table 30 shows the best TR and simulated mission time associated with the best replication of each DROP-CFA run in Instance 4.

Table 30: Problem Instance 4 DROP-CFA Superlative Results

Scenario-Run	Quadrants	Tiles	ρ^{max}	ETR	Mission Time
7-1	8	4	60	1135	59.58 min
7-2	16	4	60	1244	59.91 min
7-3	8	8	60	1135	59.58 min
7-4	16	8	60	1244	59.91 min
8-1	8	4	120	1515	118.68 min
8-2	16	4	120	1515	119.22 min
8-3	8	8	120	1515	118.68 min
8-4	16	8	120	1515	119.22 min

Figure 8 displays the difference between the DROP policy and both DROP-CFA ADP approaches' performances for both scenarios of Instance 4.

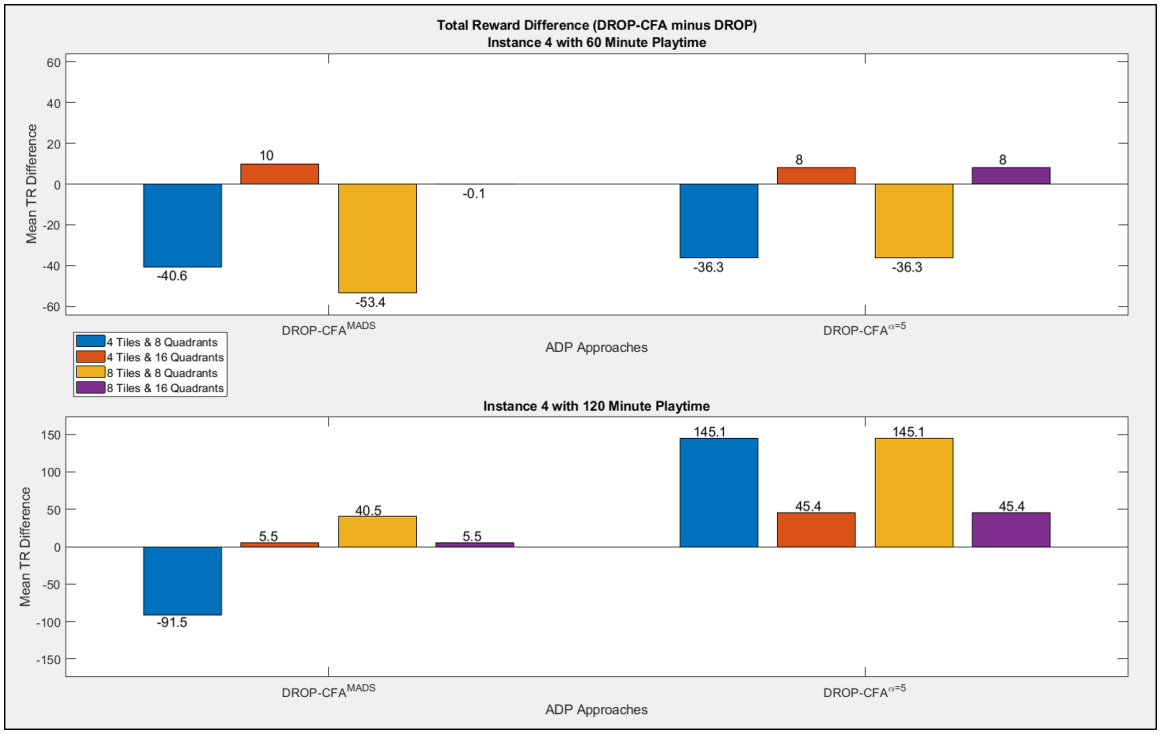

Figure 8: Instance 4 Results

For Instance 4 Scenario 7 the DROP-CFA approach with 16 quadrants as well as DROP-CFA^{MADS} approach with 4 tiles and 16 quadrants performs slightly better than the DROP policy. For Instance 4 Scenario 8, both ADP approaches perform better than the DROP policy except for one run of the DROP-CFA^{MADS} approach with 4 tiles and 8 quadrants.

4.4.6 Performance

Table 31 and Figure 9 display the best mean TR performance across all 8 scenarios, for the DROP policy, and the two ADP approaches.

Table 31: Overall Superlative Results

Scenario	Approach	Quadrants	Tiles	θ-policy	mean TR	Run Time
1	DROP-CFA^{MADS}	16	8	Eq.53	786.10±125.57	522.49 min
2	DROP	-	-	-	1043.9±118.2	45.28 min
3	DROP-CFA	16	8	Eq.59	749.3±116.63	22.64 min
4	DROP-CFA^{MADS}	16	4	Eq.58	955.80±123.04	315.62 min
5	DROP	-	-	-	902±132.41	32.41 min
6	DROP-CFA^{MADS}	8	4	Eq.62	1182.90±169.55	923.58 min
7	DROP-CFA^{MADS}	16	4	Eq.65	813.10±188.37	1447.59 min
8	DROP-CFA	8	4	Eq.68	967.7±265.29	187.27 min

The DROP-CFA^{MADS} approach performs better than the DROP policy and DROP-CFA approach in Scenarios 1, 4, 6, and 7. For Scenario 4, the DROP-CFA^{MADS} approach with 16 quadrants results in a mean TR substantially higher than the DROP policy. As discussed before, Scenario 4 in Instance 2 has dynamic targets; SAM batteries arrive uniformly across the AOR; and the AUCAV has a maximum playtime of 120 minutes. Due to the variability the SAM battery arrivals cause in each replication's ETR performance as playtime increases, creating a policy that encourages the AUCAV to travel to regions of the AOR that have high value targets before the adversary deploys a SAM battery near them is essential. This is supported by the results displayed in Figure 9.

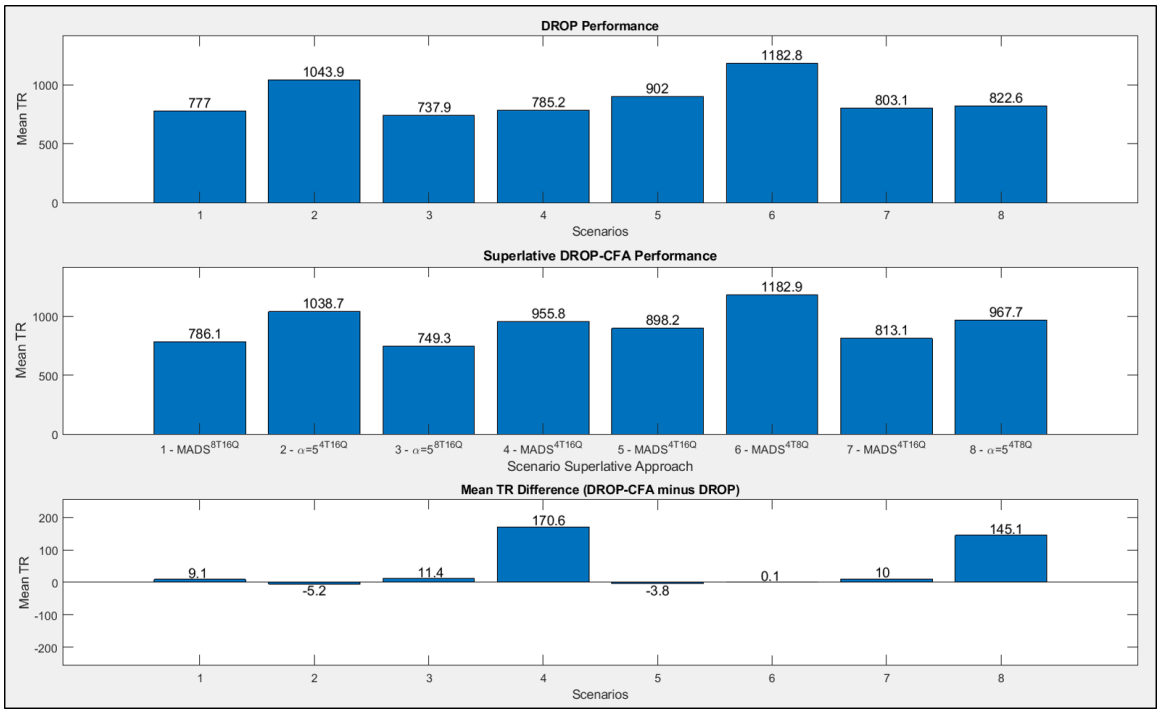

Figure 9: DROP-CFA & DROP Policy Comparison

The DROP-CFA approach performs better than the DROP policy and DROP-CFA^{MADS} approach in Scenarios 3 and 8. For Scenario 8, the DROP-CFA approach with 4 tiles and 8 quadrants results in a mean TR 145.1 points higher than the DROP policy. Scenario 8 is the exact same as Scenario 4 except the targets and SAM batteries arrive in the northeast corner of the AOR region, and the AUCAV has a maximum playtime of 120 minutes. Again, the same concept applies that encouraging the AUCAV to travel to the northeast region of the AOR, where the majority of the targets are arriving before the adversary deploys a SAM battery near them, is essential.

4.4.7 Robustness

Considering the DROP policy 95% confidence interval half-widths in Table 13 and Table 31, the half-width for the DROP-CFA^{MADS} Scenario 4 is 123.04. For the

DROP policy, it is 468.3. The DROP-CFA^{MADS} approach resulted in a significantly lower variability in ETR performance than the DROP policy. The half-widths for the DROP-CFA in Scenarios 3 and 8 are 116.63 and 265.29, respectively. For the DROP policy it is 135.68 and 430.63, respectively. The DROP-CFA approach resulted in a significantly lower variability in ETR performance than the DROP policy for Scenario 8 and slightly lower in Scenario 3. Overall, the remaining scenario half-widths are similar across the DROP and superlative runs of each DROP-CFA Approach.

4.4.8 Computational Effort

Although the DROP-CFA^{MADS} approach performs better in four out of eight of the scenarios, its use of the ORTHOMADS algorithm results in a computational effort significantly higher than the DROP policy and DROP-CFA approach. The DROP-CFA approach and DROP policy run times are similar to each other and are both significantly lower than the DROP-CFA^{MADS} approach cumulative run times. As a result, the DROP-CFA Algorithm 3 is recommended moving forward with sampling or running a design of experiment with the initial θ-policy changing the value of α, maximum tiles, and maximum quadrants. If the MILP formulation (referenced in Section 4.2.1) is replaced with a less computationally expensive formulation, the DROP-CFA^{MADS} Algorithm 2 is recommended for further exploration of the ORTHOMADS algorithm's impact on computational effort when it experiences a less computationally expensive embedded formulation. Given its high run time, the DROP-CFA^{MADS} Algorithm 2 can be considered as an offline ADP algorithm that provides a superlative θ-policy prior to a SCAR mission based on provided intelligence information. Then it's superlative θ-policy can be inputted into the DROP-CFA Algorithm 2 to gain further insight on the AUCAV's predicted performance in its next SCAR mission.

V. Conclusion

In conclusion, a policy implementing 16 quadrants for scenarios regarding the arrival of targets (i.e., Instances 1 and 3) performs better than 8 quadrants. The number of time tiles for a maximum playtime of 60 and 120 minutes does not make a significant difference for the scenarios in Instances 1 and 3. The DROP-CFA approach has the potential to perform significantly better than the DROP policy in a scenario where SAM batteries arrive and targets arrive in clusters. For the DROP-CFA, if the AOR is partitioned into quadrants that are equivalent to the size of the cluster of targets and SAM battery arrivals, the more likely it will perform better than the DROP policy.

5.1 Future Work

A design of experiment or sampling approach exploring the change in values of α, maximum time tiles, and maximum quadrants with Algorithm 3 is recommended. This would allow for a higher number of replications without the computational expense the ORTHOMADS algorithm imposes. If the ORTHOMADS algorithm is used, finding a substitute for the MILP formulation used (reference Equations 46-52) is recommended. Instead of an MILP formulation, a relaxed linear programming formulation with limited branching and bounding could be explored. Also, an linear programming formulation could be removed all together and another formulation can be implemented along with the tile and coarse coding established in this thesis.

The trends in this thesis revealed how significant the number of quadrants are on the mean TR. The number and size of quadrants for all instances at a playtime of 120 should be explored more. In addition to this, the battle-space being partitioned into more than 16 quadrants (i.e., 24 quadrants) should also be explored in future

research to gain insight on the relationship between the size of the battle-space and the number of partitioned quadrants. Lastly, the DROP-CFA approach where $\alpha = \rho_t$ without the θ^{MAD} vector should be examined to see its impact on the overall mean TR. Then, a new DROP-CFA$^{pseudo-MADS}$ approach should be implemented utilizing Algorithm 3 with $\alpha = \rho_t$ (or $\alpha = \rho_t^2$) and the θ^{MAD} vector to explore its impact on the AUCAV's mission performance and mean TR received.

Appendix A. Acronyms

ADP - approximate dynamic programming

AI - artificial intelligence

AMSTE - Affordable Moving Surface Target Engagement

AOR - area of responsibility

AUCAV - autonomous unmanned combat aerial vehicle

CFA - cost function approximate

C.I. - confidence interval

COCOM - Combatant Command

CONUS - The continental United States

CSAR - combat survival and recovery

DLA - direct lookahead approximations

DoD - Department of Defense

DROP - deterministic repeated orienteering problem

DROP-CFA - DROP policy with CFA approach

DROP-CFA^{MADS} - DROP policy with CFA approach & ORTHOMADS algorithm

EI - expected improvement

ETR - expected total reward

HPT - high priority target

HVT - high value target

IFTU - in-flight target update

JDAM - Joint Direct Attack Munition

JFC - Joint Forces Commander

JISE - Joint Intelligence Support Elements

JSTARS - Joint Surveillance Target Attack Radar System

JTF - Joint Task Force

KG - knowledge gradient

MADS - mesh adaptive direct search

MCTS - Monte Carlo tree search

MDP - Markov decision process

MILP - mixed integer linear program

NAI - named area of interest

NFZ - no-fly zone

OCONUS - Outside the Continental United States

ORTHOMADS - orthogonal mesh adaptive direct search

OT&E - operational tests and evaluations

OTN - obstacle transition node

OTR - obstacle threat ring

PFA - policy function approximation

RL - reinforcement learning

SAM - surface-to-air missile

SCAR - trike coordination and reconnaissance

SDVRP - stochastic dynamic vehicle routing problem

SEAD - Suppression of Enemy Air Defense

SecDef - Secretary of Defense

TOP - team orienteering problem

TR - total reward

TST - time sensitive target

UAV - unmanned aerial vehicle

UCAV -Unmanned Combat Aerial Vehicle

US - United States

USAF - US Air Force

USSOCOM - US Special Operations Command

VFA - value function approximations

Bibliography

Abramson, M. A., C. Audet, J. E. Dennis Jr, and S. L. Digabel (2009). OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. *SIAM Journal on Optimization 20*(2), 948–966.

Blum, A., S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff (2007). Approximation algorithms for orienteering and discounted-reward tsp. *SIAM Journal on Computing 37*(2), 653 – 670.

Butler, A. and L. M. Colarusso (2002). As uavs take hits in service poms... osd pushing air force, navy to merge ucav development projects. *Inside the Air Force 13*(41), 1–7.

Campbell, A. M., M. Gendreau, and B. W. Thomas (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research 186*(1), 61–81.

Department of Defense (2018a, Jan). *Joint Publication 3-33: Joint task Force Headquarters.*

Department of Defense (2018b, Sep). *Joint Targeting*, Volume Joint Publication 3-60 Targeting.

Department of Defense (2021, May 28,). The Department of Defense Releases the President's Fiscal Year 2022 Defense Budget. Available: `https://www.defense.gov/Newsroom/Releases/Release/Article/2638711/the-department-of-defense-releases-the-presidents-fiscal-year-2022-defense-budg/`.

Department of the United States Air Force (2019, Mar). *Operations*, Volume Air Force Doctrine Publication 3-60 Targeting.

Frazier, P. I., W. B. Powell, and S. Dayanik (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization 47*(5), 2410–2439.

Goodwill, J. C. (2021). The Autonomous Attack Aviation Problem. pp. 46–55.

Gunawan, A., H. C. Lau, and P. Vansteenwegen (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research 255*(2), 315–332.

Jansen, N., B. Könighofer, S. Junges, and R. Bloem (2018). Shielded decision-making in mdps. *arXiv preprint arXiv:1807.06096*.

Jiang, D. R., L. Al-Kanj, and W. B. Powell (2020). Optimistic Monte Carlo Tree Search with Sampled Information Relaxation Dual Bounds. *Operations Research 68*(6), 1678–1697.

Karas, R. S. (2017). As Contested Battlespace Grows, MQ-9 Explores New Roles. *Inside the Air Force 28*(26), 7–9.

Military Advantage (2014, Feb). F-35B Lightning II. Available: `https://www.military.com/equipment/f-35b-lightning-ii`.

Mizokami, K. (2021, May 28,). "The New F-15EX Scored Some Kills in Its First Big Wargame". Popular Mechanics. Available: `https://www.msn.com/en-us/news/world/the-new-f-15ex-scored-some-kills-in-its-first-big-wargame/ar-AAKtPb6`.

Murali, S. (2018). Reinforcement learning for a hunter and prey robot.

Office of the Under Secretary of Defense (Comptroller)/Chief Financial Officer (2021, May 28,). Fiscal Year 2022 Budget Request. United States Department of Defense

Budget Overview. Available: https://comptroller.defense.gov/Portals/45/Documents/defbudget/FY2022/FY2022_Budget_Request.pdf.

Pang, Z.-J., R.-Z. Liu, Z.-Y. Meng, Y. Zhang, Y. Yu, and T. Lu (2019). On Reinforcement Learning for Full-length Game of Starcraft. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 33, pp. 4691–4698.

Powell, W. B. (2019). A Unified Framework for Stochastic Optimization. *European Journal of Operational Research 275*(3), 795–821.

Powell, W. B. (2021). Reinforcement learning and stochastic optimization.

Ragi, S. and E. K. Chong (2013). UAV Path Planning In A Dynamic Environment via Partially Observable Markov Decision Process. *IEEE Transactions on Aerospace and Electronic Systems 49*(4), 2397–2412. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6621824.

Shu, Z., J. W. Ohlmann, and B. W. Thomas (2018). Dynamic orienteering on a network of queues. *Transportation Science 52*(3), 691 – 706.

Souidi, M. E. H., P. Songhao, and L. Guo (2017). Mobile agents path planning based on an extension of bug-algorithms and applied to the pursuit-evasion game. *Web Intelligence (2405-6456) 15*(4), 325 – 334.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika 25*(3/4), 285–294.

Thue, D. and V. Bulitko (2012). Procedural game adaptation: Framing experience management as changing an mdp. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Volume 8.

Ulmer, M. W., J. C. Goodson, D. C. Mattfeld, and B. W. Thomas (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics 9*(2), 100008.

Ulmer, M. W., N. Soeffker, and D. C. Mattfeld (2018). Value function approximation for dynamic multi-period vehicle routing. *European Journal of Operational Research 269*(3), 883–899.

Zheng, J. and A. Siami Namin (2018). A markov decision process to determine optimal policies in moving target. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 2321–2323.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2022	Master's Thesis	Sept 2020 — Mar 2022

4. TITLE AND SUBTITLE

Approximate Dynamic Programming for an Unmanned Aerial Vehicle Routing Problem with Obstacles and Stochastic Target Arrivals

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gurnell, Kassie, M., Capt, USAF

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-22-M-134

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

14. ABSTRACT

The United States Air Force is investing in artificial intelligence (AI) to speed analysis in efforts to modernize the use of autonomous unmanned combat aerial vehicles (AUCAVs) in strike coordination and reconnaissance (SCAR) missions. This research examines an AUCAV's ability to execute target strikes and provide reconnaissance in a SCAR mission. An orienteering problem is formulated as an Markov decision process (MDP) model wherein a single AUCAV must optimize its target route to aid in eliminating time-sensitive targets and collect imagery of requested named areas of interest while evading surface-to-air missile (SAM) battery threats imposed as obstacles. The AUCAV adjusts its route depending on the arrival locations of the SAM batteries and targets into the battle-space. An approximate dynamic programming (ADP) solution approach is developed wherein mathematical programming techniques are utilized with a cost function approximate (CFA) policy to develop high quality AUCAV routing policies to improve SCAR mission performance. The CFA policy is compared to a deterministic repeated orienteering problem (DROP) benchmark policy across four instances that explores varied arrival behaviors of dynamic targets and SAM batteries. Overall, the proposed CFA policies perform nearly the same or better than the DROP policy in all four instances.

15. SUBJECT TERMS

Markov decision process (MDP), approximate dynamic programming (ADP), vehicle routing problem (VRP), cost function approximation (CFA), mesh adaptive direct search (MADS)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Matthew Robbins, AFIT/ENS
U	U	U	UU	102	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4606; matthew.robbins@afit.edu