3-2022

Multiagent Routing Problem with Dynamic Target Arrivals Solved via Approximate Dynamic Programming

Andrew E. Mogan

Multiagent Routing Problem with Dynamic Target Arrivals Solved via Approximate Dynamic Programming

THESIS

Andrew E. Mogan, 2d Lt, USAF

AFIT-ENS-MS-22-M-156

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-22-M-156

MULTIAGENT ROUTING PROBLEM WITH DYNAMIC TARGET ARRIVALS SOLVED VIA APPROXIMATE DYNAMIC PROGRAMMING

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

Andrew E. Mogan, BS

2d Lt, USAF

March 24, 2022

AFIT-ENS-MS-22-M-156

MULTIAGENT ROUTING PROBLEM WITH DYNAMIC TARGET ARRIVALS SOLVED VIA APPROXIMATE DYNAMIC PROGRAMMING

THESIS

Andrew E. Mogan, BS
2d Lt, USAF

Committee Membership:

Dr. Matthew J. Robbins
Chair

Dr. Brian J. Lunday
Member

AFIT-ENS-MS-22-M-156

Abstract

The United States Air Force (USAF) continues to invest in the research and development of technologies leveraging artificial intelligence to produce competitive attack behavior via autonomous unmanned aerial vehicles (AUAVs). The employment of multiple AUAVs can be used as a force multiplier to assure air superiority against the enemy and remove an endangerment to the warfighter. We formulate and solve the multiagent routing problem with dynamic target arrivals (MRP-DTA), a stochastic system wherein a team of AUAVs executes a strike coordination and reconnaissance (SCAR) mission against a notional adversary. Dynamic target arrivals that occur during the mission present the team of AUAVs with a sequential decision-making process, which we model via a Markov Decision Process (MDP). The high dimensionality and continuous nature of the state space renders classical dynamic programming techniques computationally intractable. To combat the curse of dimensionality, we construct and implement a hybrid approximate dynamic programming (ADP) algorithmic framework that employs a parametric cost function approximation (CFA) and a direct lookahead (DLA) model. We utilize a mesh adaptive direct search (MADS) algorithm to tune our CFA-DLA parameterization and produce high-quality attack policies for the team of AUAVs. To demonstrate the merit of our algorithmic approach, we design an experiment to test our solution approach on multiple instances of the MRP-DTA. We compare superlative ADP policies against competitive benchmark policies; the recommended ADP policies exhibit a statistically significant improvement over the repeated greedy marginal heuristic benchmark policy for 19 of 20 problem instances tested and a statistically significant improvement over the repeated sequential orienteering problem benchmark policy for 8 of 10 problem instances tested.

We show that the probability of a high payoff target arrival and the regions in which targets arrive are critical problem features that influence the quality of the resulting policy. Results of excursional analysis show the value tradeoff of balancing solution quality and computational effort when selecting the base optimization model for our CFA-DLA algorithm.

I would like to dedicate this work to my mother, who worked incessantly to ensure I had the opportunity to write this document. Thanks for maintaining faith in my capabilities.

Acknowledgements

First, I would like to express my genuine gratitude to my advisor, Dr. Matthew J. Robbins, for his hardwork, mentorship, and genuine care for my education. I would also like to thank Dr. Brian J. Lunday for his unwavering dedication to student success and the aid he provided on this thesis. Lastly, I want to thank my family for their constant support.

Andrew E. Mogan

Table of Contents

Page

List of Figures

List of Tables

MULTIAGENT ROUTING PROBLEM WITH DYNAMIC TARGET ARRIVALS SOLVED VIA APPROXIMATE DYNAMIC PROGRAMMING

I. Introduction

The ongoing development of autonomous systems and robotic technologies presents the potential for the advancement and discovery of warfighting techniques that benefit the United States Air Force (USAF). The USAF continues to face the challenge of advancing science and technology because peer and near-peer geopolitical competitors contest the key components of its power projection (Wilson, 2019; Department of the Air Force, 2021). The USAF recognizes that new technologies such as artificial intelligence, autonomous systems, and robotics will ensure it can fight and win wars in the future (Mattis, 2018; Department of the Air Force, 2019d).

Friendly and opposing forces have begun interfacing Autonomous Unmanned Aerial Vehicles (AUAVs) with manned forces to achieve military objectives and maintain air superiority. Turkish forces recently employed such autonomous aircraft against Syrian forces during Operation Spring Shield, showing that their AUAVs could actively maneuver and attrit various military targets to include air defense systems, howitzers, and military bases before the deployment of manned assets (Haider, 2019). The USAF could benefit from the application of autonomous aircraft as a potential force multiplier in highly repetitive, dangerous operations (Cahoon, 2021). Autonomous aircraft have proven their utility in hazardous environments and can maneuver at flight regimes (e.g., acceleration forces, altitudes) not suitable for manned aircraft. Operations research (OR) methods can be applied to the field of autonomous systems to directly benefit the USAF in the development of future autonomous tech-

nologies, tactics, and procedures needed to maintain a competitive advantage.

The USAF seeks to maintain air superiority so as to permit the conduct of military operations without threat of interference from adversarial combatant forces. Historically, air superiority has been an integral perquisite to success for an operation or campaign (Department of Defense, 2017b). USAF Chief of Staff General Charles Brown emphasizes the need to localize air superiority and enable joint effects as an integral component to the security of the United States (Brown Jr, 2020). Friendly combatant forces work to establish air superiority first, so subsequent operations are executed without interference from other hostile forces (Department of the Air Force, 2019a).

After establishing air superiority, the USAF must maintain control of the fight through offensive attack operations that degrade the enemy's ability to mobilize and fight back. A primary mission used to accomplish this goal is air interdiction. The USAF primarily defines *air interdiction* as a multi-faceted effort to divert, disrupt, delay, or destroy the enemy's military potential before it can be brought to bear effectively against friendly forces or to otherwise achieve the joint force commander's (JFC's) objectives. Air interdiction must contend with many hostile conditions within the environment, requiring timely and accurate intelligence reports to inform decision-makers about the enemy's capabilities, disposition, and intentions (Meilinger, 2014). The USAF deems *counterland operations* as a critical mission set used to accomplish air interdiction.

The USAF executes counterland operations to interdict and destroy enemy ground targets dispersed across an area of operation. The JFC focuses air-to-ground attacks on key enemy targets to degrade the capabilities of the enemy and accomplish a tailored set of mission objectives throughout the area of operation (Department of the Air Force, 2020). A mission set the USAF uses in conjunction with other services

to maximize effective destruction of enemy assets is known as the strike coordination and reconnaissance (SCAR) mission.

The SCAR mission is a derivative mission associated with counterland operations that bolsters air interdiction mission objectives. During a SCAR mission, the USAF collects intelligence, surveillance, and reconnaissance (ISR) information on potential enemy targets and directs attack assets to strike targets, detect additional targets, and provide battle damage assessments (BDA) for future operations. AUAVs provide the potential for increased aircraft endurance, reduced risk to the warfighter, and superior targeting selection policies when compared to manned assets, representing a potential asset for the SCAR mission. The USAF utilizes a logically structured targeting process that allows for intelligence management by the JFC. As presented by Brunson (2007), the USAF relies solely on the JFC's objectives to establish the priority for attacking targets or supporting reconnaissance efforts. Intelligence forces obtain ISR reports and establish deliberate targets prior to the deployment of assets in the attack domain. We reference the attack domain as the two dimensional ground space over which targets are located and the temporal domain over which the SCAR mission occurs. Targets are classified based on a multitude of characteristics: the time-sensitivity of the attack window, the value of destruction, and the degradation imposed on the enemy force (Department of Defense, 2017c). Due to these varying factors, the JFC recognizes different priority levels among targets. High-payoff targets (HPTs) are targets recognized as essential targets to achieve the JFC's primary objectives. The JFC establishes a joint integrated prioritized target list (JIPTL) that explicitly describes the target service sequence. It is essential to understand that the JIPTL is usually constructed based on the value of each target; however, target position, target value, and target terrain characteristics complicate the prioritization of targets on the list and require further scrutiny. In addition to servicing HPTs, the

SCAR mission focuses on targets known as named areas of interest (NAIs). NAIs are added to the JIPTL to contribute to future intelligence on target locations and provide the JFC with BDA, which may inform future attack missions. Realistically, it is routine for targets to arrive (i.e., be identified) as the SCAR mission progresses, in real-time.

A dynamic target describes the identification of a new target after the deployment of assets (Department of the Air Force, 2019c). Although dynamic targets are identified after all deliberate targets have been prioritized, they are still viable targets if they meet the JFC's objectives. Attack assets can still service dynamic targets due to the flexibility of the targeting process; however, the arrival of these dynamic targets can sometimes change the execution of the JIPTL and thus influence the efficiency of operations. The stochasticity associated with the arrival of a dynamic target represents the primary source of uncertainty in our problem. Although it may seem optimal for assets to enter the attack domain to execute a planned SCAR mission and adjust in stride to address dynamic target arrivals, we believe that, by anticipating the arrival of dynamic targets in the attack domain, we can develop high-quality, multiagent attack policies that will outperform reactive, deterministic policies via the employment of reinforcement learning techniques. However, the introduction of multiple attack aircraft in joint airspace requires additional problem constraints to mimic proper airspace control.

The USAF uses the term *airspace control* to define the employment of multiple assets operating in a joint attack domain. Airspace control is extraordinarily dynamic and situational, but to optimize airspace use, control should accommodate users with varied technical capabilities. The necessity for airspace control is motivated by the threat level, the available surveillance, the navigation, and the technical communication capabilities of both the airspace users and the controlling agencies. These

capabilities directly inform development of coordination measures (Department of the Air Force, 2019b), which are necessary to deconflict the airspace and ensure the safe and efficient conduct of operations in accordance with (IAW) the JFC's objectives.

This research presents the multiagent routing problem with dynamic target arrivals (MRP-DTA), focusing on directing multiple autonomous attack agents in a given attack domain. The mission objective is to employ a team of AUAVs on a SCAR mission to service targets. The primary goal of an AUAV is to earn the highest total reward, where an AUAV earns a reward from servicing a target. The team of AUAVs must adjust to the arrival of dynamic targets and properly maintain airspace control through different communication scenarios. Specifically, AUAV communication with each other can be crucial to establish a productive prioritization of targets for the team of AUAVs to attack. To best represent this scenario, we use modeling techniques that properly represent stochasticity and management of communication between AUAVs.

In this research, we model the MRP-DTA using a Markov decision process (MDP) framework and attain solutions using approximate dynamic programming (ADP) techniques. The MDP framework provides a structured formulation for defining a multitude of problem features. The MDP framework models stochasticity present in real-world systems. Exact algorithms can then be applied to solve the MDP model to optimality. However, these solution techniques are not computationally tractable for the MRP-DTA due to the large size of the problem. The innumerable state and outcome spaces of the problem require a powerful approximate technique such as ADP to provide high-quality policies that better inform decision-makers in the routing of attack assets in the attack domain. We represent the MRP-DTA in a two-dimensional attack domain. The team of AUAVs enter the attack domain in search of ground targets to destroy with the knowledge of deliberate targets contained in the JIPTL. The

JIPTL enables determination of an initial, static, optimal route for each AUAV in the absence of dynamic targets. Whereas the JFC may require the deliberate targets to be attacked first, we investigate the performance of the AUAVs when given selected knowledge that characterizes dynamic target arrivals in various portions of the attack domain. The deterministic attack policies present the motivating research question; we believe these may be improved upon by capitalizing on the known probability of dynamic target arrivals throughout the attack domain. We implement a designed computational experiment to test the sensitivity of problem features and their effects on policy performance.

The subsequent material presented in this thesis follows a logical presentation of necessary problem information. Chapter 2 provides an in-depth literature review of similar problem classes, similar modeling frameworks, and applicable solution methodologies. Chapter 3 explicitly defines the problem description, the MDP model formulation of the MRP-DTA, and the ADP solution methodologies used to solve the model. Chapter 4 presents the results, targeted analysis, and gathered insights from this analysis. Finally, Chapter 5 provides recommendations for extending this research.

II. Literature Review

This chapter contextualizes the various academic fields relating to this research, giving an in-depth review of each field's contribution to the formulation of the MRP-DTA. This chapter contains five separate sections. The first section introduces the orienteering problem (OP) as a broad problem class and subdivides the OP into distinct sub-classes. The second section examines the stochastic dynamic vehicle routing problem (SDVRP) and discusses how this problem class manages the arrival of new information via dynamic target arrivals. The third section focuses on the MDP modeling framework used to capture and represent the uncertainty in our problem. The fourth section showcases the practicality of the ADP solution approach and reviews distinct ADP algorithmic designs used in similar problems. The final section describes a cooperative game theory solution approach used to appraise the value of communicative properties between multiple agents in the system.

2.1 Orienteering Problem

In the traditional OP, an agent attempts to visit as many target locations as possible before visiting the exit (i.e., departure) node, subject to a strict travel budget limiting the agent's ability to visit all nodes. The OP has been proven to be deterministically solvable using optimization techniques. As initially introduced by Golden *et al.* (1987), the distance and travel times between nodes are assumed to be known by the agent. This assumption allows for the employment of a deterministically optimal target selection policy where target selection describes the combinatorial optimization problem addressed by the agent in the system. The primary distinction in the OP and other routing problem classes exists in its objective function. The agent's objective is to maximize the total collected reward, accumulated from the agent route, wherein

the first visit to each node (target) prior to the exit node accrues a node specific award. The agent must manage its travel resource (e.g., fuel, battery life) and select a subset of targets to service. Additional constraints limit the agent from visiting targets multiple times and force the agent to start and end in predetermined locations (Vansteenwegen & Gunawan, 2019). Given the relevance of the OP class, subsequent discussion expounds upon literature that focuses on the OP and OP variants that directly relate to our problem formulation. We also discuss the subclasses of the OP to review parameters that are relevant to the formulation of the MRP-DTA.

When applied to targeting processes, the OP makes rather strong assumptions to solve the system to optimality, to include knowledge of all targets, knowledge of all service times, and knowledge of all travel times. Since 1987, researchers have introduced more realistic components into these formulations by incorporating uncertainty in the system (Papapanagiotou *et al.*, 2015; Thayer & Carpin, 2020, 2021), teams of agents (Chao *et al.*, 1996; Panadero *et al.*, 2017; Bayliss *et al.*, 2020), or other restrictive components, making the OP problem class more robust in its application to various target selection problems.

2.1.1 Physical Orienteering Problem

The physical orienteering problem (POP) is relevant to routing AUAV assets in a military application. The POP presents a problem sub-class of the OP wherein the agent has prior knowledge of obstacle locations within the target domain of the agent and must navigate a coordinate grid while avoiding these obstacles. Air defense systems serve as the primary deterrent of unauthorized aerial access to a region, providing obstacles in the form of circular areas that must be avoided. In the interest of protecting valuable assets, air defense systems are typically dispersed strategically throughout an area (Department of the Air Force, 2019a). When planning and

executing a route, the agent must determine a collision-free route that maximizes the expected total reward collected over the budgeted travel limit. The additional complexity of avoiding obstacles adds constraints to the OP formulation, increasing the computational effort required to solve the routing problem to optimality. High-quality solutions have been derived using a search metaheuristic wherein the algorithm continually drives toward an improved solution with a limit on computational effort (Pěnička *et al.*, 2019). These results support future operations that require the routing of autonomous, unmanned assets around hostile air defense systems.

2.1.2 Orienteering Problem with Replenishment

Wallace *et al.* (2020) present a variant of the OP known as the orienteering problem with replenishment (OPR). The OPR allocates onboard organic munitions given to the agent, which adds an additional resource constraint to the OP. The agent can replenish its onboard resource at specified charging nodes, referred to as recharging stations. The agent must manage its given travel budget and munitions throughout the time horizon. The management of an additional resource adds complexity to the problem by increasing the decision space, which is accompanied by additional constraints. As the problem parameters are expanded, the OPR suffers from the curse of dimensionality. The OPR must remain a small problem with a low number of maximum revisits to scale the problem to a tractable level. Given a small problem instance, the results suggest that autonomous agents may extend past their primary utilization as an ISR asset. In the fight to exhaust the enemy, these replenishment sites can increase the lethality of US forces and increase the efficiency of attack assets downrange.

2.1.3 Team Orienteering Problem

A common variant of the OP involves the team orienteering problem (TOP) first introduced by Chao *et al.* (1996), which models the integration of multiple agents into the standard OP. The TOP introduces a special consideration wherein the team of agents must work together to accumulate maximum reward, with each target node being restricted to one visit by the team of agents. This restriction follows many real-world applications and adds a level of complexity when attempting to solve the problem. To manage the use of multiple agents, the problem formulation must track the status of each agent in the system at all times.

Jeong *et al.* (2014) investigate a basic level formulation of a multiple agent targeting selection problem in a surveillance application. The problem considers the employment of multiple UAVs over an infinite horizon. The reward associated with visiting an unsurveilled area may grow. The MDP model formulation scales the problem to a tractable level. The objective seeks to maximize the expected total discounted reward over an infinite horizon, which motivates optimal steady-state behavior. The authors solve for an optimal policy that manages all UAVs in the system. To better articulate the uncertainty in the system, the authors leverage Shannon's Entropy (Jeong *et al.*, 2014). The simulation presents a small, restrictive version of the MRP-DTA by only including five available nodes. When applied to more extensive problems with a larger number of target nodes, the TOP becomes computationally demanding.

The approach taken by Panadero *et al.* (2017) shows the power of simulation optimization in dealing with both the stochasticity and team dynamic in the TOP with stochastic travel times and service times. A sim-heuristic allows for robust routing of agents in the system to better solve the TOP with uncertainty compared to deterministic solution methods, which require large amounts of computational effort. Bayliss

et al. (2020) apply a solution approach that sacrifices solution quality for less computational effort. The authors incorporate realistic travel constraints into their model, which introduces uncertainty into the travel time between targets. The authors use a machine learning mechanism that approximates the cost of traversing each edge, given the motion constraints. The successful integration of a heuristic to solve the TOP can be a powerful tool in providing high-quality solutions to the TOP with a significantly lower computational burden.

2.1.4 Stochastic Orienteering Problem

The stochastic orienteering problem (SOP) is an OP variant that includes at least one of several different problem features involving uncertainty. Uncertainty can manifest in the agent's travel times, the agent's target service times, or the agent's rewards gained for servicing targets. The uncertainty in the system provides the potential to use an MDP modeling framework to find an optimal policy that best prescribes the optimal actions of the agent throughout the time horizon.

Thayer & Carpin (2020) model the SOP with stochastic travel times and service times. The servicing agent traverses a network of grapevines to service various irrigation nodes located throughout the vineyard. Given external factors and difficulties, the agent is unable to deterministically predict the time needed to travel to each serviceable target in the system or the time required to service each target. An MDP model captures the uncertainty in the system and allows the system to be solved to optimality, providing the agent with an optimal travel policy. The model also considers the agent's value tradeoff of collecting more reward at the expense of failing to return to the destination node. The model incorporates a failure probability, allowing the agent to assume a certain level of risk to travel longer and accumulate additional reward.

Thayer & Carpin (2021) extend this concept by focusing on an adaptive path policy algorithm that determines policies using specified deterministic paths as a starting point and branching, if necessary. In this work, the authors leverage a branch heuristic to reduce the computational cost of using this algorithm and display results accordingly. It is common to see this problem framework in applications dealing with customer service wherein the service provider must make a pre-established obligation to serve a set of customers without either knowledge of the travel times to reach customers or service times with the customers (Papapanagiotou *et al.*, 2015). Such application areas formulate a penalty delegated to the agent if the service provider is unable to meet the predetermined obligation because the agent not only incurs zero rewards but effectively incurs a cost of failed service. Although this application area requires the specification of a predetermined path to solve to optimality, other application areas benefit from dynamic re-routing, allowing for an agent to strategically abandon a path if a new target arrives and may be optimal to service.

2.2 Stochastic Dynamic Vehicle Routing Problem

The SDVRP presents a like problem class to the standard OP wherein geographical targets are arranged throughout a domain and require service via one (or, in our case, any of more than one) traveler while experiencing dynamic target arrivals after the agent begins to traverse their planned route. As the name implies, the SDVRP experiences stochastic target arrivals over the decision-making horizon. This new information provides the dispatching authority with potential adjustments for routing assets, which requires a robust modeling framework and high-power computational abilities to identify high-quality rerouting policies. Differences arise between the SDVRP and the OP in the objective function. The SDVRP prioritizes visiting all customer nodes while minimizing the cost accrued by the agent, whereas the OP

prioritizes maximizing expected total reward subject to a route distance constraint.

The research on SDVRPs is vast and spans many application areas and multiple solution approaches. The fundamental dilemma in solving the SDVRP to optimality depends on the agent's resiliency to new information. Pillac *et al.* (2013) review reoptimization techniques used to adjust the vehicle's routing instructions after starting its route. Reoptimization is an event-driven technique in which the optimal vehicle route is recalculated after new information arrives in the system. Reoptimization can be computationally expensive, creating a fundamental tradeoff for problem solvers as the return in total reward may not be worth the effort required to achieve it.

The static counterpart of the vehicle routing problem requires a less reactive approach that implements an *a priori* policy that cannot be adjusted during the execution of the route. The appeal of a static approach can be motivated by lower computational cost or lack of access to vehicle positional data that allows for the adjustment to routing (Pillac *et al.*, 2013). On the contrary, adjusting route plans according to the arrival of new targets almost always guarantees a superior routing policy. Modern dynamic programming techniques have proven robust in solving the SDVRP via the utilization of the MDP modeling framework.

The incorporation of route planning into the MDP model by Ulmer *et al.* (2020) provides an intuitive appeal to solving the SDVRP denoted as the route-based MDP model. This representation emulates the framework used for routing problems but increases the problem's state space dimensionality because the system must now manage all available routes at any decision epoch. Ulmer *et al.* (2020) contend the increased dimensionality is worthwhile because one attains the superior ability to express solution methods in terms of the prescribed model elements. This ability is especially integral to incorporating multiple agents working in tandem because the framework can be leveraged to show route planning information for each agent in the system.

The fundamental challenge when dealing with multiple agents is ensuring each agent acts optimally regarding other agents in the system. Sundar *et al.* (2016) present multiple variants of the formulation for the multiple vehicle routing problem (MVRP) wherein a team of vehicles must visit a set of nodes while minimizing the cost of travel. The variants of the MVRP highlight the different communicative properties available when modeling the MRP-DTA.

2.3 Markov Decision Processes

The MDP modeling framework is a robust modeling framework used across many different application areas to solve for optimal policies and inform decision-makers. The MDP modeling framework describes all aspects of a given system by explicitly defining the decision epochs of the system, state space of the system, action space available given the state of the system, transition functions, and reward functions (Puterman, 1994).

The MDP modeling framework is touted for its superior ability to represent stochasticity in complex systems that exhibit a sequential decision making process. Hubmann *et al.* (2017) develop an MDP model to investigate decision policies in Advanced Driver Assistance Systems (ADAS). In this problem, car manufacturers are working toward the production of various ADAS to support autonomous driving. These systems retrieve data from their environment and make sequential decisions intended to result in safe and logical transportation policies. An online MDP models the uncertainty of different traffic participants and generates the optimal behavior. This formulation informs our research because the vehicle exhibits anticipatory behavior in an environment of uncertainty. The vehicle must choose the acceleration/deceleration policy that maximizes the expected total discounted reward as the vehicle travels through various traffic scenarios. The model is tested under three different

traffic scenarios of increasing complexity, accurately predicting other vehicles' actions and responding accordingly. The optimal policy displays intelligent behavior by decelerating the car to better assess the other vehicle's actions.

The use of the MDP model is prevalent in problems where the agent wishes to anticipate the actions or location of other vehicles. Li *et al.* (2019) model a single UAV moving through an airspace. The UAV is subject to collision with other intruder aircraft in its airspace. The UAV can either use an automated decision-making process to avoid the intruding aircraft or rely on a human pilot to conduct an avoidance maneuver. The work specifically focuses on the uncertainty associated with communication latency to construct an MDP that allows the UAV to act optimally in a given situation and avoid collision with an intruder aircraft. The authors define the objective of the MDP model to determine the optimal waiting strategy for the UAV. The results show that the optimal waiting time depends on the position and velocity between the UAV and the intruder aircraft as well as the intruder's motion model. The authors use value iteration to obtain the optimal stationary policy. The MDP model creates a map of optimal wait times that can be stored in the UAV's memory and referenced in future scenarios. The UAV can reference these mappings to determine the optimal wait time for a pilot's command when encountering intruder vehicles. The results show the ability to derive optimal policies in applications relating to autonomous vehicles and further support the MDP model framework as a valuable technique for modeling uncertainty in a system.

2.4 Approximate Dynamic Programming

This section showcases the value of ADP solution techniques in solving computationally intractable, large-scale MDP model formulations. The dimensionality required to represent complex systems renders exact solution algorithms intractable.

Large-scale Markov decision problems require the implementation of approximation techniques to combat the curse of dimensionality (Powell, 2011). In our case, the system suffers from an uncountable state space that requires the use of approximation techniques to develop high-quality routing policies for the team of AUAVs. Our work leverages the use of the TOP mathematical formulation to improve upon competitive benchmark policies and solve for a high-quality routing policy. We review the value function approximation (VFA) algorithmic approach, direct lookahead approximation (DLA) approach, and the cost function approximation (CFA) approach for solving large scale sequential decision problems.

2.4.1 Value Function Approximation

A VFA approach approximates the value of occupying the current state when explicitly computing the value function for all state-action pairs is too large a computational burden. VFA requires less computational effort because it more efficiently represents the value of occupying each state of the system at a given time by iteratively sampling and estimating the value of a subset of states.

VFA has been a critical component in the MEDEVAC literature by developing high-quality ADP policies that outperform the myopic dispatch policy used in practice (Jenkins *et al.*, 2021b,a; Rettke *et al.*, 2016; Robbins *et al.*, 2020). In military MEDEVAC operations, dispatchers utilize a closest-available policy that dispatches the closest available MEDEVAC unit to a casualty. In high-intensity combat operations, ADP policies have been shown to outperform the closest available policy. Due to the high-dimensionality of the MEDEVAC MDP model, the development of an ADP algorithm provides a computationally tractable solution for dispatchers.

Least-squares temporal differences (LSTD) is a linear architecture that utilizes a set of basis functions to approximate the value function for a fixed policy. Ret-

tke *et al.* (2016) derive high-quality solutions using LSTD within a policy iteration algorithmic framework, conducting a designed experiment to tune algorithm performance. A 3^3 full factorial design is performed on the arrival rate and algorithmic parameters. Results show the ADP policy outperforming the myopic policy across all experimental levels. Jenkins *et al.* (2021b) add to the MEDEVAC literature by similarly modeling the MEDEVAC system using an MDP model and further solving the system using a support vector regression (SVR) VFA within a policy iteration algorithmic framework. The formulated ADP algorithm derives a high-quality dispatch policy that outperforms the closest available dispatch policy as the arrival rate of casualties increases. Extending that contribution, Jenkins *et al.* (2021a) expand the model formulation by incorporating the redeployment of MEDEVAC assets into the MDP model. This helps build a more realistic formulation that allows a dispatching authority to task a MEDEVAC unit to a service request before it returns to its original staging area, assuming that it can refuel and reequip at the current Medical Transport Facility (MTF). Jenkins *et al.* (2021a) show that a combination of techniques can be paired together by using both LSTD and neural network learning to evaluate different candidate policies. Using both techniques yields high-quality policies that outperform the currently accepted closest-available policy.

VFA has proven to be invaluable for solving systems that involve routing assets tasked with servicing target nodes. The use of VFA in the ADP framework provides substantial results in military MEDEVAC research and informs the construction of our ADP algorithm.

2.4.2 Direct Lookahead Approximation

When dealing with dynamic target arrivals or stochastic systems, deterministic solution techniques ignore stochastic information and execute what is optimal, given

the literal target information available to the agent in the current decision state. DLA policies take a more practical approach to dealing with uncertainty in problem classes wherein uncertainty manifests in future system states. Techniques such as horizon truncation permit the agent to rollout the horizon of the problem far enough to capture system changes relevant to the current decision. This suggests that the length of the horizon can be used to tune algorithm performance where longer horizons are generally desired if computationally appropriate. These rollout algorithms have been proven to outperform deterministic-based benchmark policies in various applications of the OP (Zhang *et al.*, 2018). Rollout policies demonstrate their usefulness in problems with complex interactions between resources in the system. The rollout policy requires the availability of a base policy that initializes the policy for the agent. After establishing the base policy, the rollout algorithm uses specified decision points throughout the agent's horizon to adjust the agent's policy. Encouraging predictive behavior via DLA policies improves upon a posteriori solutions when stochastic demands are present (Secomandi, 2001). Secomandi (2001) recommend the integration of multiple agents as an improvement of their proposed methods.

The combination of VFA and online rollout algorithms has been shown to provide high-quality routing policies that inform dispatching authorities on the optimal routing of assets in response to dynamic target arrivals. Ulmer *et al.* (2019) develop an offline-online ADP algorithm wherein their offline VFA leverages spatial and temporal elements of the post-decision state and combines this with an online rollout algorithm to achieve anticipatory behavior. Furthermore, the results show that the temporal information yields a better estimation of the reward-to-go than spatial information. Using both spatial and temporal information is of paramount importance to developing our ADP framework to solve for high-quality routing solutions.

2.5 Cost Function Approximation

Lookahead approximation approaches can be computationally demanding, given the large enumeration of potential outcomes when applied to complex stochastic systems. Particularly, the use of a lookahead policy (i.e., a deterministic model) is evaluated via simulation of the system to estimate the value of that policy but requires numerous calls to solve the model and can produce less than desirable results. A common approach to improving the quality of these lookahead policies is via parametric CFA wherein the embedded deterministic model of the lookahead policy is modified via a parameterization of problem parameters (Ghadimi *et al.*, 2020). This approach searches various policies via either a gradient based or gradient-free search technique to produce an improved policy. The parameterization is applied to either the objective function of the base optimization model or the constraints of the base optimization model.

Implementing a CFA algorithm requires two primary focuses for a stochastic system: developing the parameterization of the base optimization model and solving for the superlative parameterization to the model. In the application of a CFA with a deterministic model, we consider the approach to be a CFA-DLA hybrid approach due to the forecasting of future decisions using the deterministic model (DLA component) and the parameterization of the base optimization model (CFA component). Perkins & Powell (2017) are the first to show the utility of applying a CFA in a simulation based framework. Their solution approach avoids the complex and computationally demanding issue of calculating lookahead approximations and tunes a stochastic base model via simulation optimization. Their parameterization approach properly manages the complex dynamics of the stochastic system and yields improved results over their base optimization model. Perkins & Powell (2017) parameterize an energy storage problem to show an improvement over the basic deterministic looka-

head model, directly demonstrating the success of applying CFA approach to the energy storage problem class. Similarly, Shuai *et al.* (2019) studies a energy management problem wherein the constraints of the optimization model are parameterized to better consider the effects of stochasticity in wind power. A stochastic gradient descent algorithm is used to tune the model parameters and solve for a high-quality scheduling solution. The field of CFA based solution approaches is understudied in the literature, although commonly used in industry to induce high-quality solutions, which motivates future research into the power of the CFA strategy when applied to sequential decision problems.

2.6 Cooperative Game Theory

The integration of multiple agents in an attack domain presents various approaches to managing communications between agents. Given a potential threat to the communication structure between AUAVs, the study of non-communicative formulations informs stakeholders on the worst case performance for the team of AUAVs. Each agent must coordinate with other agents to share route planning information throughout the attack mission to truly act optimally. It may be necessary to involve an omniscient planner that aids in route planning and deconfliction of the two agents; however, full autonomy between agents may be desired to avoid communication breakdowns imposed by the enemy. In the absence of communication between agents, the system can be modeled using a cooperative game theory approach. The formulation treats each agent as a player attempting to maximize their utility. Each player may or may not cooperate with others to improve their collective and individual utilities. Communication is known to produce a better targeting selection policy; however, Thakoor *et al.* (2019) theoretically prove that a complete lack of communication between agents in the system results in at most half the expected reward of the op-

timal solution obtained when relying on communication between agents. The study of such radio silenced formulations supports the value of communicative properties when employing multiple AUAVs in military attack operations and warrants further investigation into policy performance under various communication scenarios.

III. Methodology

3.1 Problem Definition

In this section, we elaborate on current military operations and strategies necessary to frame the MRP-DTA. We specifically describe the role of air interdiction and offensive counterland operations to contextualize the goal of developing a high-quality policy to route a team of autonomous attack assets. We then examine the current USAF targeting policy and the process for categorizing a deliberate or dynamic target. Next, we present the framework for the SCAR mission. Finally, we present the MRP-DTA and define the mission for the team of AUAVs.

3.1.1 Air Interdiction

Air interdiction is the primary mission used to render the enemy's capabilities ineffective against friendly forces (Department of the Air Force, 2020). Air interdiction is fundamental to achieving air superiority and is a primary mission for attack aviation assets executing counterair and counterland operations. Counterair operations encompass attack operations focused on degrading enemy airpower, which may include attacking aircraft, missiles, or anti-aircraft defense systems (Department of the Air Force, 2019a). The primary goal of counterair operations is to establish air superiority. The enemy uses defense systems to protect air assets, making any counterair mission a highly dynamic and situational decision making process. The use of autonomous attack assets to conduct counterair operations requires avoidance behavior and is a potential, related research endeavor.

Counterland operations use attack aviation assets to destroy and disrupt enemy land force capabilities to achieve JFC objectives (Department of the Air Force, 2020). Counterland missions are integral to exhausting the enemy's resources and winning

the battle. Targets deliberately scheduled for destruction in counterland operations include enemy strongholds and infrastructure, logistic systems, communication nodes, and attack assets (Department of the Air Force, 2020). Each target provides a potential benefit to achieving JFC objectives in a region; thus, each target varies in potential reward for destruction.

The successful conduct of counterland operations requires joint capabilities from other United States military branches (Department of Defense, 2019b). The JFC establishes fire support coordination measures (FSCM) to properly integrate forces and avoid potential fratricide in the joint area of operations (AO). Establishing proper FSCMs includes defining three critical boundaries on the battlefield: the forward line of troops (FLOT), the fire support coordination line (FSCL), and kill boxes. Figure 1 provides a notional joint AO that explicitly defines all boundaries on the battlefield. The FLOT lies at the forward-most location where friendly troops may be located. Beyond the FLOT, operational commanders identify the FSCL to permit the conduct of joint interdiction methods. Beyond the FSCL, forces target and destroy enemy assets without having to deconflict operations with ground troops. For this reason, operations beyond the FSCL promote speedy and deadly attack operations. Any attack operation conducted in the area contained between the FLOT and the FSCL must be coordinated with the proper land force commander. This restriction does not mean that interdiction efforts may not be conducted in this region. However, when operations occur beyond the FSCL, the efforts are conducted at such distance from friendly forces that detailed integration of each air mission with the fire and movement of friendly forces is not required (Department of the Air Force, 2020). For this reason, the location of the FSCL should strike a balance so as not to unduly dampen the operational tempo of ground forces while maximizing the effectiveness of organic and joint force interdiction assets (Department of Defense, 2016).

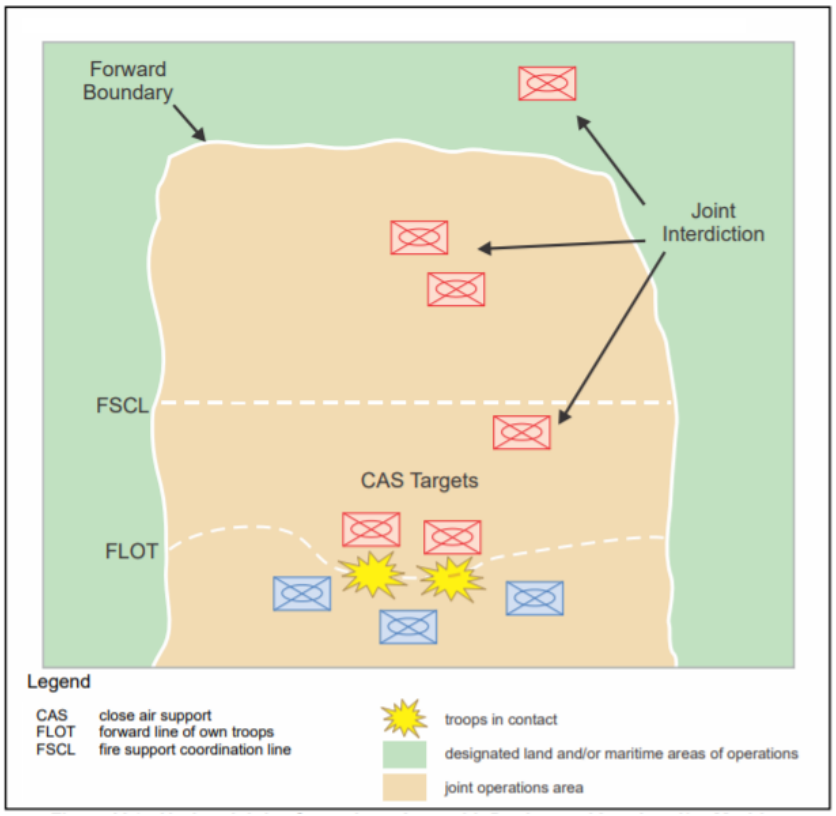

Figure 1. Notional Joint Operations Area with Designated Land Area of Operations (Department of Defense, 2016)

When deploying assets beyond the FSCL, assets will be limited due to either weapon capacity or fuel capacity. A forward arming and refueling point (FARP) constitutes a location where forces replenish weapons and fuel. FARPs provide a quick turnaround for attack assets. AUAVs utilize a FARP to increase their endurance for FO-related air interdiction activities forward of the FSCL. Future research may investigate the impact of the location of the FARP.

Autonomous attack aviation assets provide functionality as a forward observer (FO) that directs friendly indirect fire from other assets to the target. The role of an autonomous attack asset acting as a FO beyond the FSCL provides a distinct advantage when striking the enemy. By calling upon exterior sources to destroy targets, aviation assets can avoid detection by enemy forces and avoid any limitation on weapons capacity. Furthermore, autonomous attack assets may suffer from adverse weather conditions that influence the destruction of targets, whereas platforms such as the AC-130U "Spooky" gunship can easily overcome these constraints (USAF, 2021).

The AC-130U's primary mission includes close air support, air interdiction, and armed reconnaissance. The primary advantage of the AC-130U is its superior ability to strike from high altitude and avoid anti-aircraft fire. Weather conditions help inform the altitude at which the AC-130U strikes, and on occasion, the altitude required may put the aircrew at risk of enemy air defense systems. USAF leadership and other respected AC-130U aircrew attribute the combat loss of the "Spirit 03" AC-130U gunship in 1991 to an Iraqi anti-aircraft missile striking the left-wing of the aircraft while conducting operations at 9,000 ft (Hicks, 2014). The AC-130U can successfully reduce the risk of a strike from enemy air defense systems by performing operations at higher altitudes. The use of autonomous aviation acts as a force multiplier because it provides the AC-130U aircrew with the necessary sensor capabilities

to identify targets while the AC-130U maintains a safer altitude away from enemy air defense systems. Autonomous attack assets acting as FOs relay target location information back to the AC-130U and allow for external strike and destruction of targets beyond the FSCL. This distinct advantage enables relaxation of constraints within on our mathematical model because we assume that the munitions capacity of the AC-130U is large enough to service all targets identified by a team of AUAVs performing an air interdiction mission.

3.1.2 Air Force Targeting Process

The targeting process is the center of success for any air interdiction mission. Which targets to strike, when to strike targets, and how to strike targets are all decisions that ultimately affect successful air interdiction efforts. Although the primary responsibility of success lies with the JFC, the responsibility of selecting targets is typically delegated to the commander, Air Force forces (COMAFFOR), or commonly referred to as the joint force air component commander (JFACC) (Department of the Air Force, 2019c). The JFACC typically works within an air operations center (AOC), leading a staff of personnel to manage and execute the targeting process in accordance with the JFC's original guidance. On the JFACC's staff is the target effects team (TET), which connects the targets and capabilities with the potential desired effects and deconflicts and coordinates target nominations via the JIPTL. The TET is critical to the target development phase seen in Figure 2.

The target development stage is the planning stage during which the JFACC and JFACC's staff focus on deliberate targeting of targets that directly accomplish the objectives and desired effects of the JFC. The JFACC's staff consists of targeteers within the ISR division (ISRD), the combat plans division (CPD), the TET, and the non-kinetic operations coordination cell (NKOCC) (Department of the Air Force,

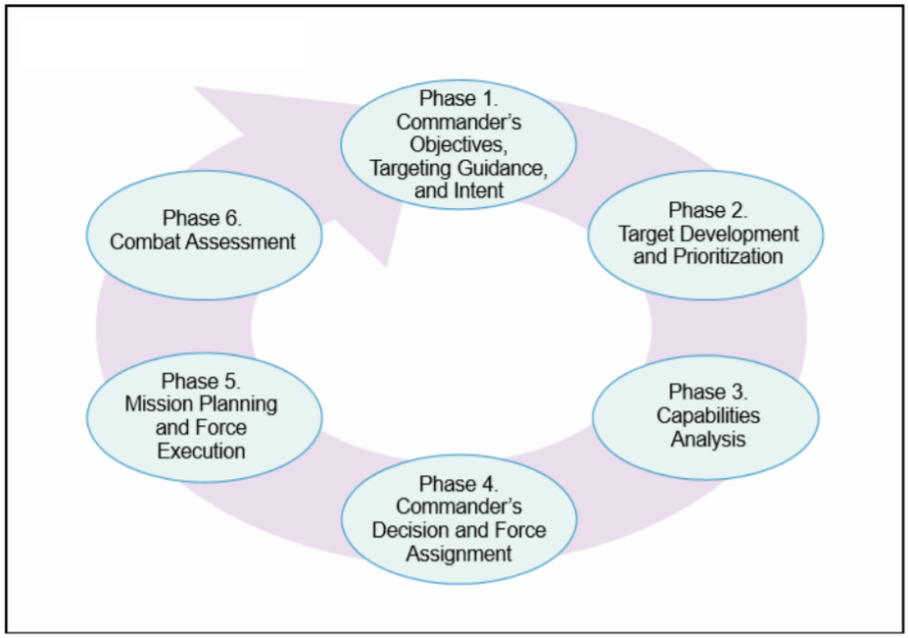

Figure 2. Joint Targeting Cycle (Department of Defense, 2019a)

2019c). This team is solely responsible for determining which targets should be struck and the sequence in which to strike them. To aid in this process, the JFACC's staff performs five distinct functions during the targeting process. These functions are target analysis, target vetting, target validation, target nomination, and identification of intelligence gaps, collection, and exploitation (IICE) requirements. These functions are summarized in Table 1.

Table 1. Five Distinct Functions for Target Development (Department of the Air Force, 2019c)

	Description
Target Analysis	Match specific targets to desired effects
Target Vetting	Asses accuracy of intel used to develop target
Target Validation	Ensure targets are compliant with law of war
Target Nomination	Nominate targets for service
IICE	Collection of data and BDA after target attack

After completing these steps, and the JIPTL has been established and further approved by the JFC, air tasking orders (ATOs) are released to the responsible execution

components. Figure 3 shows a notional JIPTL wherein the targets are scheduled for destruction by priority. The JFACC's staff scrutinizes the priority of targets using analytical targeting tools.

An analytical targeting tool is a standard criterion designed to aid the expeditious classification and scheduling of deliberate targets and dynamic targets. Given the JFC's discretion, such tools use standardized criteria to ensure proper prioritization of targets. A CARVER tool is an analytical targeting tool used to classify targets where the target is evaluated by its criticality, accessibility, recuperability, vulnerability, effect, and recognizability (Department of the Army, 2019). A CARVER tool helps to quickly classify HPTs and other lower priority targets in accordance with the JFC's objectives. An analytical targeting tool can also be pivotal to scheduling dynamic targets as the evaluation criteria for these targets are the same but their prioritization and integration into an existing JIPTL must occur at a much faster pace.

A target can be detected by intelligence forces after assets have been launched to attack deliberate targets, thus resulting in an emerging target and a potential dynamic target. Whereas emerging targets are identified during the implementation of an ATO, dynamic targets are the subset of emerging targets important enough to be serviced immediately. The flow chart in Figure 4 helps to describe the classification of an emerging target and the proper follow-on action. Although the targeting evaluation criteria is the same, the emerging target must be quickly classified and potentially scheduled for destruction if the target accomplishes mission objectives.

This research is predicated on deriving high-quality re-routing policies that leverage the information of dynamic target arrivals after the initial routing of assets according to the JIPTL. We believe these policies derived from ADP can outperform static policies that ignore this information while executing the original route plan provided via the JIPTL. The SCAR mission is both an attack and reconnaissance mission,

PRIORITY	BE# or UIC	Osfx	Cat	CC	Target	Location			Remarks	Nom by	JFACC task in priority	USER requested priority	Army track #	Previous target category nom
						Lat	Lon	Alt						
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
01					WEBB AFLD					ACC	01A3-1	01		A, B, C
02					REESE AFLD				Shut 24 Hrs	ACC	01A3-1	02		B,C
03					LUBBOCK AFLD				Shut 24 hrs	ACC	01A3-1	03		B,C
04					TERRY AFLD				Shut 24 hrs	ACC	01A3-1	04		C
05					NODONG STORAGE SITE				DESTROY	ARFOR	02W3-1	01	3E1501N	
06					SCUD CC STORAGE SITE				DESTROY	ARFOR	02W3-1	02	3E1502N	
07					SCUD B STORAGE FAC				DESTROY	ARFOR	02W3-1	03	3E1503N	
08					SAN ANGELO SA-2 SITE 4 RDR FAC					ACC	03A3-4	15		
09					AMARILLO ADEF IOC AND RADAR FAC					ACC	03A3-4	16		A
10					SAN ANGELO ADEF IOC AND RADAR FAC					ACC	03A3-4	17		A

LEGEND	
(a)	JIPTL priority
(b)	BE: a specific identification number or point location of a facility or installation
(c)	A specific identification number or point location: in conjunction with a Facility BE Number
(d)	Category code
(e)	Country code
(f)	Target Name
(g)	Location: 3 D coordinates
(h)	Location: 3 D coordinates
(i)	Location: 3 D coordinates
(j)	Desired effect
(k)	Nominator
(l)	Applicable tactical task
(m)	Nominator's priority order
(n)	Army track number
(o)	Previous targeting criticality category nominations

LEGEND (ACRONYMS)

A/C	aircraft	BE	basic encyclopedia	JIPTL	joint integrated prioritized target list
ACC	air component commander	Cat	category	Lat	latitude
ADEF	air defense (foreign)	CC	country code	Lon	longitude
AFLD	airfield	FAC	forward air controller	Nom	nominated
Alt	altitude	Hrs	hours	RDR	Radar
ARFOR	Army forces	IOC	intelligence operations center	SA	selective availability (GPS)
AW	air warfare	JFACC	joint force air component commander	UIC	unit identification code

Figure 3. Example JIPTL (Department of Defense, 2017c)

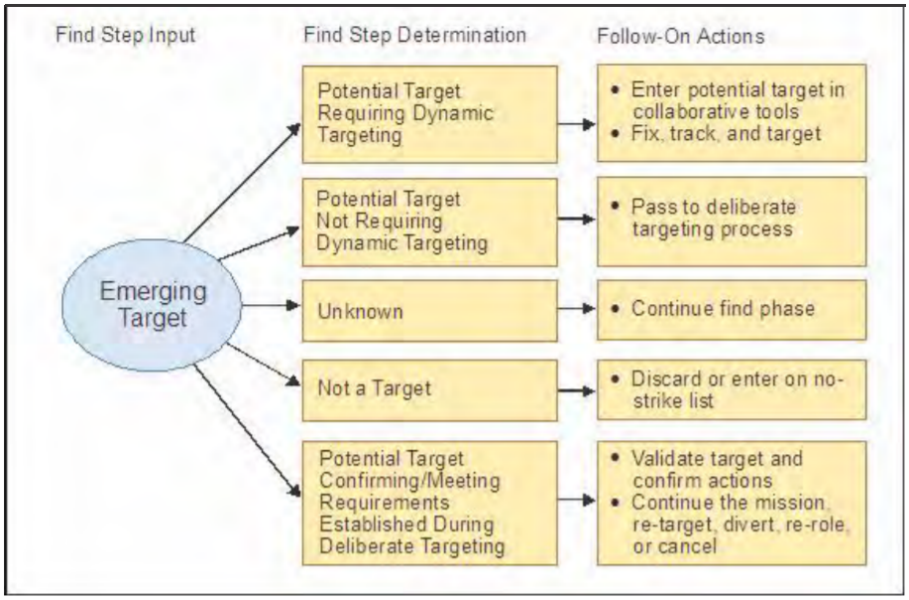

Figure 4. Find Step Determination and Actions (Department of Defense, 2017c)

providing the ability to evaluate the performance of both static and dynamic routing policies.

3.1.3 SCAR Mission

The SCAR mission focuses on hasty, dynamic targeting for air interdiction missions in a box or grid attack domain. The fundamental goals of the SCAR mission are to detect and destroy targets, neutralize enemy air defenses, and obtain BDA (Department of the Air Force, 2020). These goals inform the formulation of the MRP-DTA and the attendant MDP model.

Dynamic targeting requires adaptability by the friendly forces to successfully execute the SCAR mission. Our problem context involves a team of AUAVs deployed on a SCAR mission according to the JIPTL generated by the JFACC's staff. As each AUAV executes its planned route, new targets arrive and are evaluated using the same evaluation criteria for each static target initially scrutinized in the targeting cycle. The JIPTL is not a static document and may be adjusted after the initial

routing of assets. The SCAR mission centers on quick adaptation to these dynamic target arrivals. It requires each AUAV to quickly adapt to the arrival of new targets, to re-route and attack these targets if deemed optimal to do so. AUAVs are also motivated to perform BDA, a critical component of the targeting cycle, further accomplishing the JFC's objectives.

BDA is the process of analyzing previously struck targets to determine the functionality of the target. Specifically, BDA can "determine the degree of success in achieving objectives and to formulate any required follow-up actions, or to indicate readiness to move on to new tasks in the path to achieve overall JFC objectives" (Department of the Air Force, 2019c, p. 85). Although BDA is an element of the SCAR mission, we are interested in evaluating the performance of the team of AUAVs in the presence of dynamic target arrivals. These new targets may be located relative to local terrain features located in the attack domain.

Terrain features provide the enemy with a significant advantage and motivate the establishment of resources. Intelligence preparation of the battlefield (IPB) is a crucial element of preparing for a SCAR mission. During IPB, intelligence forces work to identify key terrain features and determine areas where intelligence collection efforts should be focused (Grindle *et al.*, 2004). The enemy forces may value a terrain feature for its concealment abilities, its difficulty to access, or superior strike abilities (Department of the Army, 2019). These areas provide enemy ground forces with defensible terrain and further motivate the discovery of target areas to hinder the enemy's ability to move and strike friendly forces. We use terrain features to inform and adequately direct intelligence collection efforts to best discover dynamic targets. If deemed eligible by the JFACC and JFACC's staff, these dynamic targets may be scheduled for service by the team of AUAVs. The MRP-DTA is formulated based on the principles of the SCAR mission.

3.1.4 The MRP-DTA

The MRP-DTA models a mission wherein two AUAVs embark on a SCAR mission to interdict targets and advance JFC objectives given a standardized analytical targeting tool and initial JIPTL. The team of AUAVs performs a functional role as a FO for the AC-130U strike platform that officially interdicts a target after an AUAV has visited and confirmed the target. Afterward, each AUAV performs the role of BDA and proceeds to the next target. The JFACC and JFACC's staff provide the tasking orders and JIPTL that initially route the team of AUAVs. As the AUAVs traverse the region, intelligence forces are working to process dynamic target arrivals across the geographical attack domain with a higher focus on key terrain features that provide the higher potential for new target arrivals. At the discovery of dynamic targets, the team of AUAVs will either adjust to the arrival of information or continue on the pre-mission route previously assigned to the team. The team of AUAVs is motivated to confirm and destroy targets available across the geographical attack region, given a uniform fuel budget for each AUAV. After each AUAV has completed its planned route, it must return to the FARP located at the same point of initial deployment. Although each AUAV is fully autonomous, each operates via communication with a human dispatching authority.

The dispatching authority is known as a "human on the loop" and has full control to halt the execution of each AUAV's target engagement. In the case where communications are severed with the dispatching authority, an AUAV will immediately return to the FARP and cease target engagement (Sayler, 2020; Department of Defense, 2017a). The dispatching authority primarily exists to monitor the behavior of the AUAVs rather than physically control their behavior throughout the mission.

Each AUAV can visit three different types of targets: NAIs, regular payoff targets (RPTs), and HPTs. By visiting and collecting intelligence information at an NAI, the

JFC is better informed on enemy composition, disposition, or terrain composition, providing crucial information to future targeting decisions (Department of Defense, 2020). Furthermore, each AUAV services either RPTs or HPTs to gain rewards. The reward for servicing an RPT, HPT, or NAI is a preference parameter that depends upon the JFC's mission objectives. For example, if the JFC prioritizes the attack mission over the reconnaissance mission, the reward for servicing both RPTs and HPTs will exceed that of visiting a NAI. The destruction of an HPT inherently earns a higher reward than the destruction of an RPT, no matter the relative prioritization of the reconnaissance mission or attack mission.

Dynamic route planning policies are implemented, assessed for total destructive impact, and compared against policies generated by our high-quality ADP algorithms. A dynamic routing policy refers to a policy that reactively adjusts the AUAV route to the arrival of new targets during the SCAR mission. These dynamic policies are used as benchmark policies to quantify the quality of our ADP policies. Although these benchmark policies provide high-quality results, an ADP algorithm that proactively adjusts to the arrival of new targets through anticipatory behavior should provide the JFC with superior destructive results.

In the next section, we formulate the MDP model for the MRP-DTA. Given the sequential decision-making process of servicing dynamic target arrivals, an MDP model is a suitable model formulation.

3.2 MDP Model

This section introduces the mathematical formulation of the MRP-DTA as an infinite-horizon MDP model. Mathematical models perform an intermediary role between real-world applications and prescriptive solution methods. Specifically, the use of our mathematical model allows us to concisely state the problem and further solve

it while considering all complex interactions in the system (Ulmer *et al.*, 2020).

MDP models are used to model sequential decision-making processes via a framework wherein the following model components are defined: decision epochs, state space, action space, transition functions, and contribution functions (also referred to as the reward function). The objective of solving a model to optimality is to determine the best route plan for the team of AUAVs given the initial layout of HPTs, RPTs, and NAIs (succinctly referred to as targets in subsequent sections) in the attack domain. The model is solved subject to each AUAV's fuel capacity constraint. Each AUAV is motivated to obtain the highest expected total reward by servicing a set of targets. As new information arrives, each AUAV is presented with a set of decisions given the current state of the system that may change the initial route plan. In practice, exact dynamic programming techniques such as backwards induction, value iteration, and policy iteration can solve a tractable MDP model to optimality providing the dispatching authority with an optimal policy. An optimal policy provides a decision-maker with an optimal action given any state of the system. If followed, this policy provides the decision-maker with the best possible sequence of decisions. However, these exact approaches are computationally demanding, especially when the dimensionality of the state variable and decision variable become large in application. In our case, the MDP model provides the basis for developing a powerful ADP algorithm that produces a high-quality routing policy. Table 2 references key notation and acronyms needed to formulate the MDP model. We model the problem using common vehicle routing terminology where AUAVs are considered entities or agents in the system model.

Table 2. Key Acronyms and Notation for MDP Model Formulation

Acronym	**Description**
HPT	High payoff target
RPT	Regular payoff target
NAI	Named area of interest
TAI	Target area of interest
JIPTL	Joint integrated prioritized target list
Notation	**Description**
λ	Target arrival rate
$\mathcal{T}$	Set of all decision epochs
S_t	State of the system at epoch t
P_t	Physical state variable at epoch t
$\ell_t^{A^1}$	Two-dimensional location of AUAV 1 at epoch t
$\ell_t^{A^2}$	Two-dimensional location of AUAV 2 at epoch t
ρ_t	Playtime remaining at epoch t
g_t	Agent indicator variable at epoch t
$\mathcal{M}_t$	Set of all targets in the attack domain at epoch t
M_{tm}	Target status tuple at epoch t
ℓ_m	Two-dimensional target location at epoch t
ξ_m	Priority of target m
y_{tm}	Target status of target m at epoch t
τ	Current system time
e	Current event type
U	TAI status tuple
ℓ_u	Location of TAI u
$\mathcal{X}_{S_t}$	Set of all actions given S_t
x_{t1}	Action for AUAV 1 given S_t
x_{t2}	Action for AUAV 2 given S_t
Ω	Two-dimensional location of the exit node
Δ_τ	Change in system time
κ	Fixed speed of both AUAVs
$h_t^{A^1}$	Heading of AUAV 1 at time t
$h_t^{A^2}$	Heading of AUAV 2 at time t
r^{HPT}	Reward gained by servicing a HPT
r^{RPT}	Reward gained by servicing a RPT
r^{NAI}	Reward gained by visiting a NAI
$-Z$	Cost administered if an AUAV fails to return to the FARP
W_t	Exogenous information realized at epoch t

3.2.1 Decision Epochs

During the IPB phase, the JIPTL is approved and disseminated to the dispatching authority. The JIPTL explicitly provides all locations of targets and further tracks the status of each target. AUAVs start their mission and realize new information at the same moments during the SCAR mission. We formulate the MDP model as a continuous-time MDP model wherein the system is driven by incoming events that change the available information. All events that drive the progression of the system are seen in Table 3. We consider the target assignment and routing of a team of two AUAVs.

Table 3. MRP-DTA Event Types

Event	Description
1	AUAV 1 services/visits a HPT/RPT/NAI/TAI (target serviced)
2	AUAV 2 services/visits a HPT/RPT/NAI/TAI (target serviced)
3	New target realized and added to the JIPTL (target arrival)

At the occurrence of an event, new information is realized, and the system must be re-evaluated. Figure 5 illustrates the inter-event process by which the team of AUAVs realizes new information, processes this information, acts on the information, and collects reward. The inter-event process is nested in the six-step targeting process: find, fix, track, target, engage, and assess (F2T2EA). F2T2EA applies equally to the use of military capabilities to achieve lethal or nonlethal effects through non-kinetic means, such as information operations, airdrop, space operations, or directed energy (Department of the Air Force, 2019c). Triggering events drive the system's evolution. We denote one decision period as the random time between events, directly implying that each decision period is not fixed in duration. The source of randomness lies in the stochastic arrival of dynamic targets across the attack domain after the start of the mission, which we assume occurs according to a Poisson Process. Furthermore,

we can then calculate the inter-arrival time of targets to the JIPTL in accordance with an exponential distribution with rate λ. A target that has arrived to the system is added to the JIPTL and subsequently available for destruction. Although three dif-

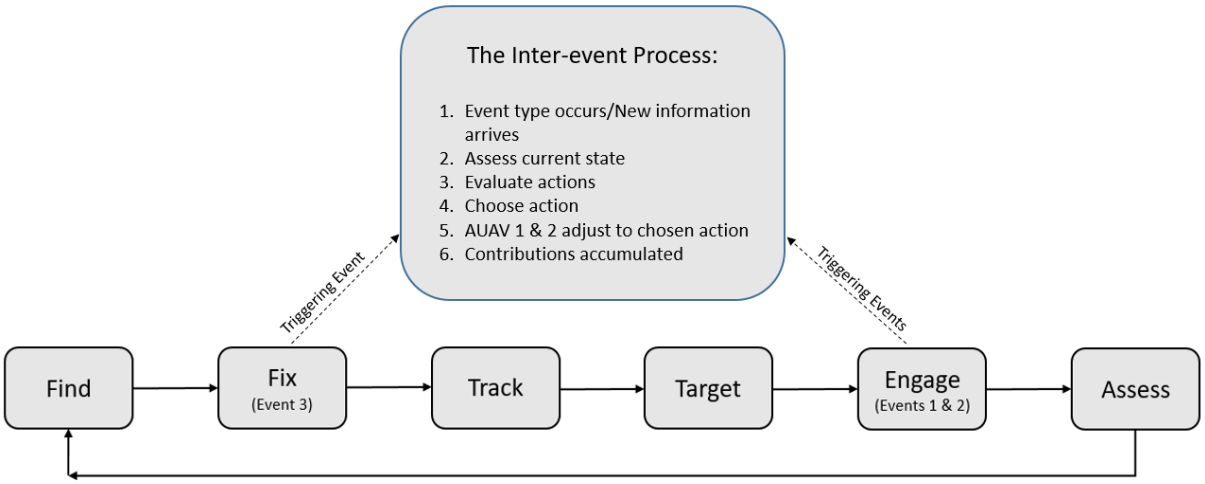

Figure 5. The Inter-Event Process

ferent event types drive the system transition, event type 3 is the only event type that introduces new information to the system. This is relevant to the decision-making process because we choose to only recalculate the route plan for the team after a new target arrival. The team of AUAVs adjusts to this new targeting information based on both the established optimality criteria and constraints of the problem.

Due to an infinite number of potential events occurring during the horizon of the SCAR mission (regardless of playtime), the MRP-DTA is formulated as an infinite horizon problem. We denote $\mathcal{T} = \{0, 1, 2, ...\}$ as the set of decision epochs by which the dispatching authority implements and manages route-plan decisions. The team of AUAVs starts at the FARP (both the start and exit node) and initializes the system with an initial route in reference to the JIPTL produced by intelligence forces. The initial route explicitly identifies the first target for each AUAV. As the mission commences, the system progresses when events trigger the system. When both AUAVs return to the exit node, the system enters the terminal state at which the team of

AUAVs experiences no cost or reward. This type of problem formulation is known as an episodic task wherein the entities execute their task over an infinite horizon and enter a terminal state at which they perpetually exist. To properly solve the MRP-DTA, we must make some deterministic assumptions regarding each AUAVs' behavior.

Consider the following assumptions regarding the MRP-DTA. Any RPT or HPT physically reached by an AUAV (exact coordinates of the target equals the exact coordinates of an AUAV) results in confirmed strike and destruction of the target via the supporting AC-130U. We assume any NAI physically reached by an AUAV (exact coordinates of the NAI equals the exact coordinates of an AUAV) results in required intelligence attained via the sensors onboard the AUAV. Any target serviced or NAI visited is immediately removed from the attack domain and may not be engaged again. The team of AUAVs deploy from the start node (the FARP), which also acts as the departure node (sometimes referred to as the exit node). The team is highly encouraged to return to the departure node before expending all fuel (playtime). At the moment an AUAV visits the departure node (exact coordinates of the departure node equals the exact coordinates of an AUAV), the AUAV has finished its mission and is forbidden to leave the departure node. An AUAV cannot be destroyed via enemy strike, mid-air collision, or malfunction. There is no assumed capacity on munitions for the AC130U. Finally, we assume that both AUAVs travel at a constant speed and expend fuel at a uniform rate.

3.2.2 State Variable

The state variable S_t captures all information necessary to model the system from t onwards and consists of information needed for computing contributions, decisions, and system transitions (Powell, 2022). The MRP requires a state representation of

the location of each AUAV, the remaining playtime allotted for the team, the status of all available targets, the current system time, and the current event type. We follow the proposed representation from Powell (2011) where P_t represents the physical state variable. The physical state variable holds critical information regarding the physical state of the system and provides information needed to transition the system or determine contributions. We denote

$$P_t = (\ell_t^{A^1}, \ell_t^{A^2}, \rho_t, g_t)$$

as the team status tuple (alternatively referred to as the physical state variable). The team status tuple holds the location information for each AUAV, the playtime remaining for the team of AUAVs, and the number of active AUAVs in the system. Let $\ell_t^{A^1}$, $\ell_t^{A^2} \in \mathbb{R}^2$ denote the location of each AUAV in the attack domain and ρ_t the playtime remaining for the team of AUAVs. We reference $g_t \in \{0, 1, 2, 3\}$ as the indicator variable distinguishing whether both AUAVs are active in the system ($g_t = 3$), AUAV 2 is only active in the system ($g_t = 2$), AUAV 1 is only active in the system ($g_t = 1$), or neither AUAV is active in the system ($g_t = 0$).

We face the challenge of representing and maintaining all targeting information realized by the team of AUAVs at epoch t. We construct the target status tuple to describe all relevant targeting information at epoch t to include each target's location, type, and status. Let $\mathcal{M}_t$ be defined as the set of all targets at time t which mathematically represents our JIPTL. We define $M_t = (M_{tm})_{m \in \mathcal{M}_t}$ as the target status tuple at time t and denote the status of each target as

$$M_{tm} = (\ell_m, \xi_m, y_{tm}), m \in \mathcal{M}_t$$

where $\ell_m \in \mathbb{R}^2$ represents the location of a target $m \in \mathcal{M}_t$ (e.g., NAI, RPT, HPT)

within the defined attack domain. Let $\xi_m \in \{0, 1, 2\}$ denote target $m \in \mathcal{M}_t$ type where 0 denotes an RPT requesting destruction, 1 denotes an HPT requesting destruction, and 2 represents an NAI requesting visitation. The variable $y_{tm} \in \{0, 1\}$ denotes target $m \in \mathcal{M}_t$ status at epoch t where 0 represents a target has been serviced or visited and 1 otherwise. Once a target m is added to the JIPTL, its location and type remain fixed for the remainder of the mission.

We leverage target areas of interest (TAIs) to facilitate mission success and promote anticipatory behavior for the team of AUAVs. A TAI is a geographical area where high-value targets can be acquired and engaged by friendly forces (Department of Defense, 2021). Each TAI is a fixed location in the attack domain that provides zero value in destruction and thus is unlikely to be visited by an AUAV due to zero inherent value. Although seemingly insignificant, TAIs play a critical role in our system by designating locations that encourage anticipatory behavior. The dispatching authority may allow an AUAV to route to an area with a high-probability of target arrivals. In our system, we denote the TAI status tuple as

$$U = (\ell_u)_{u \in \mathcal{U}},$$

wherein $\ell_u \in \mathbb{R}^2$ denotes the location of TAI $u \in \mathcal{U}$ in the attack domain and $\mathcal{U}$ denotes the set of all TAIs. Note that their fixed locations and unchanging nature allows us to exclude this information from the state variable.

We compile all variables into a succinct representation of the state variable. We define the state variable as

$$S_t = (P_t, M_t, \tau, e),$$

wherein we track the physical state of the system, the target status of all targets, the current system time denoted as τ, and the current event type denoted as e.

3.2.3 Decision Variable

The dispatching authority is the primary decision-making authority for the team of AUAVs and is ultimately responsible for managing the route plan for each AUAV. Recall the inter-event process in Figure 5. A new epoch is generated at the realization of new information to the system, and the current system state must be evaluated for a new action. If a new target arrives, triggering a new decision epoch, the dispatching authority has a set of decisions at its disposal. The dispatching authority chooses the best action that results in the highest contribution for the team of AUAVs. This may result in an AUAV abandoning its route to the next target and adopting a new route in the attack domain. This type of decision making process is known as a dynamic decision making policy because it adjusts to the arrival of new information to the system.

The decision variable encompasses Steps 3, 4, and 5 from the inter-event process. The decision variable includes all information necessary to route each AUAV to a subsequent node (i.e., NAI, TAI, RPT, HPT, or the departure node). We utilize x_{t1} to represent the location of the next target to visit for AUAV 1 and x_{t2} to represent the location of the next target to visit for AUAV 2. The state-dependent decision space, denoted as $\mathcal{X}_{S_t}$, is the set of all potential actions for the team of AUAVs at epoch t given the current state of the system S_t. The dispatching authority selects an action $x_t = (x_{t1}, x_{t2}) \in \mathcal{X}_{S_t}$, which provides the new targeting information for each AUAV. Note that $\mathcal{X}_{S_t}$ is constrained by $x_{t1} \neq x_{t2}$ where both AUAVs cannot target the same target. Equation (1) represents the set containing all potential nodes that each AUAV may visit in the system given state S_t.

$$x_{t1}, x_{t2} \in \mathcal{X}_{S_t}^{single} = \{m \in \mathcal{M}_t : y_{tm} = 1\} \cup \{U\} \cup \{\Omega\}, \quad \forall \ S_t \in \mathcal{S}, t \in \mathcal{T} \qquad (1)$$

Given this defintion, we can explicitly describe the entire decision space for the MRP-DTA. Equation (2) represents the set of all available decisions given state S_t.

$$\mathcal{X}_{S_t} = \{(x_{t1}, x_{t2}) \in (\mathcal{X}_{S_t}^{single})^2 : x_{t1} \neq x_{t2}\}, \quad \forall \ S_t \in \mathcal{S}, t \in \mathcal{T} \tag{2}$$

We denote $\Omega \in \mathbb{R}^2$ as the location of the departure node. The system is assumed to enter a terminal state only when both AUAVs reach the departure node, at which point the decision at epoch t onward given state S_t may be represented by $x_t = (\Omega, \Omega)$. If only one AUAV returns to the departure node Ω, the other AUAV is permitted to continue the mission; however, the AUAV that has exited the attack domain may no longer return to the attack domain. The action space is further constrained to only include nodes that either AUAV has sufficient time to service and return to the departure node. Future research may incorporate a proper risk assessment into the model to observe the critical points at which leadership may value the destruction of targets over the return of an AUAV.

3.2.4 State Transitions

A state transition is embedded in the inter-event process and denotes the evolution of the system from S_t to S_{t+1} given exogenous information, a chosen action, and a deterministic result. The transition function explicitly describes all elements needed to transition the system to the next decision epoch properly. Stochasticity manifests in the system via dynamic target arrivals, causing the system to transition to a new state at the trigger of any event contained in Table 3. We represent the arrival of exogenous information realized at epoch t as W_{t+1}. Equation (3) denotes the system model $S^{M,e}$, by which the system transitions.

$$S_{t+1} = S^{M,e}(S_t, x_t, W_{t+1}), \tag{3}$$

The system model accepts the state S_t, a decision x_t, and exogenous information W_{t+1}, and transitions the system to a future state S_{t+1} based off of the triggering event e that occurs. We further define the process by which the system transitions according to each event type.

A state transition is triggered anytime an AUAV services or visits a target node. We represent the system transition anytime a target node is visited by AUAV 1 as

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

where the system is triggered by event type 1. Notice that no exogenous information arrives to the system; therefore, no recalculation of either AUAV route plan is required. The system then transfers to state S_{t+1}. We denote the system transition of the location of AUAV 1 as

$$\ell_{t+1}^{A^1} = x_{t1},$$

where the location of AUAV 1 is updated to the current location of the node that AUAV 1 chose to visit at epoch t. We further model the system transition of the location of AUAV 2 as

$$\ell_{t+1}^{A^2,x} = \ell_t^{A^2,x} + (\Delta_\tau \kappa cos(h_t^{A^2}))$$

and

$$\ell_{t+1}^{A^2,y} = \ell_t^{A^2,y} + (\Delta_\tau \kappa sin(h_t^{A^2}))$$

where we represent the change in the position of AUAV 2 as a function of the location at decision epoch t, the change in the system time from the previous decision epoch Δ_τ, the fixed speed for both AUAVs κ, and the cosine and sine of the heading for AUAV 2, denoted as $h_t^{A^2}$. We further adjust y_{tm} from 1 to 0 to indicate the serviced node is no longer serviceable.

We separately represent the system transition anytime a target node is visited by AUAV 2 as

$$S_{t+1} = S^{M,2}(S_t, x_t),$$

where the system is triggered by event type 2. Once again, no exogenous information arrives to the system. The system then transfers to state S_{t+1}. We denote the system transition of the location of AUAV 2 as

$$\ell_{t+1}^{A^2} = x_{t2},$$

where the location of AUAV 2 is updated to the current location of the node that AUAV 2 chose to visit at epoch t. We further model the system transition of the location of AUAV 1 as

$$\ell_{t+1}^{A^1,x} = \ell_t^{A^1,x} + (\Delta_\tau \kappa cos(h_t^{A^1}))$$

and

$$\ell_{t+1}^{A^1,y} = \ell_t^{A^1,y} + (\Delta_\tau \kappa sin(h_t^{A^1}))$$

where we represent the change in the position of AUAV 1 as a function of the location at decision epoch t, the change in the system time from the previous decision epoch Δ_τ, the fixed speed for both AUAVs κ, and the cosine and sine of the heading for AUAV 1, denoted as $h_t^{A^1}$. We further adjust y_{tm} from 1 to 0 to indicate the serviced node is no longer serviceable.

When a new target arrival occurs, a system transition is triggered from S_t to S_{t+1}. We represent the system transition anytime a target arrives to the system as

$$S_{t+1} = S^{M,3}(S_t, x_t, W_{t+1}),$$

where the system is triggered by event type 3. Event type 3 represents an event trigger where exogenous information has arrived to the system and a new path plan must be calculated. First, we denote the system transition of the location of both AUAVs by Equations (4)-(7).

$$\ell_{t+1}^{A^1,x} = \ell_t^{A^1,x} + (\Delta_\tau k cos(h_t^{A^1})) \tag{4}$$

$$\ell_{t+1}^{A^1,y} = \ell_t^{A^1,y} + (\Delta_\tau k sin(h_t^{A^1})) \tag{5}$$

$$\ell_{t+1}^{A^2,x} = \ell_t^{A^2,x} + (\Delta_\tau k cos(h_t^{A^2})) \tag{6}$$

$$\ell_{t+1}^{A^2,y} = \ell_t^{A^2,y} + (\Delta_\tau k sin(h_t^{A^2})) \tag{7}$$

We also observe an additional target added to the target set $\mathcal{M}_t$. The system model allows us to transition the system anytime a triggering event generates a new decision epoch, utilizing the transition function associated with the current event type e.

3.2.5 Contribution Function

The accumulation of contributions drives the desired behavior of the team of AUAVs. The contribution function maps each state-action pair to a reward value for the team of AUAVs. As the system time progresses, the team of AUAVs abides by an optimality criterion that aggregates contributions and computes the team's performance during a given problem instance.

The team of AUAVs is assigned reward anytime a target is serviced or visited. The reward is immediately collected upon the visitation of an AUAV to any target on the JIPTL. If the system occupies a state at which an AUAV occupies either a node without a target (occurring when a triggering event occurs transitioning the system before a AUAV reaches its selected target) or a TAI, this AUAV contributes

no reward to the aggregate team reward. Equation (8) designates the contribution function for the MRP-DTA system.

$$C(S_t, x_t) = \begin{cases} r^{NAI} \sum_{m \in \mathcal{M}_t} \mathbb{I}_{\ell_m \in \{\ell_t^{A_1}, \ell_t^{A_2}\}, \xi_m = 2, y_{tm} = 1, \rho_t > 0} \\ r^{HPT} \sum_{m \in \mathcal{M}_t} \mathbb{I}_{\ell_m \in \{\ell_t^{A_1}, \ell_t^{A_2}\}, \xi_m = 1, y_{tm} = 1, \rho_t > 0} \\ r^{RPT} \sum_{m \in \mathcal{M}_t} \mathbb{I}_{\ell_m \in \{\ell_t^{A_1}, \ell_t^{A_2}\}, \xi_m = 0, y_{tm} = 1, \rho_t > 0} \\ -Z, \text{ if } \ell_t^{A_1} \neq \Omega \text{ or } \ell_t^{A_2} \neq \Omega, \ \rho_t \leq 0 \\ 0, \qquad \text{otherwise.} \end{cases} \tag{8}$$

We represent Z as an enormous cost that outweighs the sum of all potential rewards in the system. This cost is only administered to the team of AUAVs if either AUAV does not return to the departure node before the end of the playtime, and it intrinsically motivates each AUAV to exit the attack domain before its fuel resource is exhausted.

The JFC and their supporting team should be conscious of the relative reward assigned to each target because it can dramatically change the behavior of the team of AUAVs. Specifically, the team should work to assign these values in accordance with the overall mission objectives. If any target is assigned a reward that drastically overwhelms all other targets, the team of AUAVs will act aggressively toward this target type. It is key for the JFC, supporting staff, and dispatching authority to fine-tune these model parameters to achieve the desired battlefield effects.

3.2.6 Optimality Equations

Our model directs the operation of each AUAV over the time horizon of the MRP-DTA. The objective for the MRP-DTA is to maximize the expected total reward

(ETR) seen in Equation (9).

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi} \Big[\sum_{t=1}^{\infty} C(S_t, X_t^{\pi}(S_t)) | S_0 \Big] \tag{9}$$

The objective of the MRP-DTA is to maximize the cumulative contributions accumulated for the team of AUAVs given the starting state, S_0. The system of equations we wish to solve is known as the optimality equations (sometimes referred to as the Bellman Equations). By solving for $V(S_t)$, we can derive the optimal activity for the AUAVs in the system. The result is an optimal policy. We denote

$$V(S_t) = \max_{x_t \in \mathcal{X}_{S_t}} \Big(C(S_t, x_t) + \mathbb{E}\big[V(S_{t+1}) \big| S_t, x_t \big] \Big), \tag{10}$$

as the system of optimality equations. A solution yields an optimal policy that ultimately governs the behavior for the team of AUAVs. As previously mentioned, solving the optimality equations can be computationally intractable. The continuous nature of the attack domain renders the state variable uncountable and Equation (10) computationally unattainable. The MRP-DTA is an example of an MDP model that suffers from the well-known curse of dimensionality (Powell, 2011). To overcome this issue, one may use ADP techniques to approximate Equation (10) through iterative sampling and estimation of the value function to produce high-quality policies. Ultimately, we derive an ADP policy and compare its performance against competitive baseline routing policies to showcase the effectiveness of ADP in producing high-quality routing policies for the MRP-DTA. This research focuses on applying a cost function approximation approach to augment a base optimization model via a set of parameters and produce superior results. Our benchmark policies inform the construction of our base optimization model for our ADP algorithm.

3.3 Benchmark Policies

An optimal policy is unattainable for the MRP-DTA due to a continuous state variable representation of the problem, but the notion of solving the system to optimality is desirable for ADP policy comparison. Given that it is impossible to solve the system to optimality, we need available benchmark policies to indicate the solution quality of our ADP policies. We propose three benchmark policies to gauge policy performance. These policies are the repeated team orienteering problem policy (π^{RTOP}), repeated sequential orienteering problem policy (π^{RSOP}), and the repeated greedy marginal heuristic policy (π^{RGMH}). Table 4 presents all policies with the policy type for each respective policy.

Table 4. Benchmark and ADP Policies

Notation	Description	Policy Type
π^{RTOP}	Repeated Team Orienteering Problem	Benchmark
π^{RSOP}	Repeated Sequential Orienteering Problem	Benchmark
π^{RGMH}	Repeated Greedy Marginal Heuristic	Benchmark
$\pi^{CFA-RTOP}$	Cost Function Approximation w/ RTOP policy	ADP
$\pi^{CFA-RSOP}$	Cost Function Approximation w/ RSOP policy	ADP
$\pi^{CFA-RGMH}$	Cost Function Approximation w/ RGMH policy	ADP

We denote $\pi^{CFA-RSOP}$ ($\pi^{CFA-RGMH}$) as the ADP policy using a CFA approach with the RSOP (RGMH) policy applied to the base optimization model. This is further discussed in Section 3.4.1 where we detail the base optimization model and its role in the CFA algorithm.

3.3.1 Repeated Team Orienteering Problem Policy

The first proposed benchmark policy is the repeated team orienteering problem (RTOP) policy. The RTOP policy is a dynamic policy that solves an underlying TOP instance as new information arrives to the system. Given the position of both AUAVs,

a set of target nodes, remaining playtime, and the exit node, the RTOP policy returns the optimal routing of AUAVs at that epoch. To best adjust to new target arrivals, the RTOP policy re-solves a TOP instance upon arrival of new information that triggers a system transition. Although the RTOP policy adjusts to dynamic target arrivals, the policy is not anticipatory in nature. The RTOP policy is a reactionary policy which responds to information after it arrives to the system. We are interested in showing that our ADP policy can capitalize on the stochastic information of the system and anticipate target arrivals to exhibit a higher performance when compared to the reactionary RTOP policy.

The mathematical formulation of the classical TOP explicitly defines an objective function, constraints, and decision variables. We explicitly solve the TOP formulation from Vansteenwegen & Gunawan (2019), a mixed-integer linear program (MILP). The formulation is depicted as

$$\max \quad \sum_{m=1}^{M} \sum_{i=2}^{|N|-1} P_i y_{im}, \tag{11}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} \sum_{j=2}^{|N|} x_{1jm} = \sum_{m=1}^{M} \sum_{i=1}^{|N|-1} x_{i|N|m} = M, \tag{12}$$

$$\sum_{m=1}^{M} y_{km} \leq 1; \ \forall \ k = 2, ..., (|N| - 1) \tag{13}$$

$$\sum_{i=1}^{|N|-1} x_{ikm} = \sum_{j=2}^{|N|} x_{kjm} = y_{km}; \ \forall \ k = 2, ..., (|N| - 1); \ \forall \ m = 1, ..., M \tag{14}$$

$$\sum_{i=1}^{(|N|-1)} \sum_{j=2}^{|N|} \rho_{ij} x_{ij} \leq \rho_t, \tag{15}$$

$$2 \leq u_{im} \leq |N|; \ \forall \ i = 2, ..., |N|; \ \forall \ m = 1, ..., M \tag{16}$$

$$u_{im} - u_{jm} + 1 \leq (|N| - 1)(1 - x_{ijm}); \ \forall \ i, j = 2, ..., |N|; \ \forall \ m = 1, ..., M \tag{17}$$

$$x_{ijm} \in \{0,1\}, \forall\ i,j \in N;\ \ \forall\ m \in M \tag{18}$$

wherein the objective is to maximize the total reward attained by servicing nodes in the set $N = \{1, ..., |N|\}$ given the set of manageable agents $m = \{1, ..., M\}$. P_i denotes the reward received from visiting node i. The set of nodes also includes the current location of AUAV 1 ($\ell_t^{A^1}$ as the first node in the set) and the current location of AUAV 2 ($\ell_t^{A^2}$ as the second node in the set). The exit node (Ω) is the last node in the set ($|N|$). All other nodes in the system can be explicitly described as the set of targets in the attack domain at time t represented by $i \in \{M+1, ..., (|N|-1)\}$. Herein, the decision variable x_{ijm} equals 1 if AUAV m travels directly from node i to node j, and 0 otherwise. The decision variable y_{im} equals 1 if AUAV m visits node i, and 0 otherwise. Finally, the decision variable u_{im} indicates the ordinal sequence in which AUAV m visits node i on their route. The system is constrained by Constraints (12) - (18) where Constraint (12) ensures each AUAV route begins from the starting node and ends at the departure node. Constraint (13) ensures that each node is visited at most once during the mission. Constraint (14) maintains the connectivity of each route. Constraint (15) ensures the limited time budget (ρ_t) for each route is not exceeded by summing the distance of each arc, denoted as ρ_{ij}. Constraints (16) and (17) prevent subtours from occurring in each route during the mission. Constraint (18) enforces the binary nature of the decision variables. The collection of constraints and objective function allows us to solve the system to optimality to obtain the optimal routing for the team of AUAVs.

The RTOP policy is recalculated at each new target arrival to the system. The policy outputs the optimal routing for the team given the current state of the system. The RTOP policy is a highly competitive routing policy for autonomous agents; however, implementing the RTOP policy is computationally demanding for a large

number of targets in the system. The large computational cost of solving the TOP renders the RTOP policy an infeasible solution approach for inclusion within our ADP algorithm but still provides high-quality benchmark results for ADP algorithm comparison. To lessen the computational burden while minimizing the impact on solution quality, we leverage both the RSOP and RGMH policies.

3.3.2 Repeated Sequential Orienteering Problem Policy

The RSOP policy represents a sequential routing technique wherein each AUAV is sequentially routed via solving the orienteering problem (OP). Any time a new target arrival occurs, the OP is solved for AUAV 1, the available target set ($\mathcal{M}_t$) is adjusted, and the OP is solved for AUAV 2. The RSOP policy outperforms the RTOP policy with respect to solution speed while underperforming with respect to solution quality. Although the RSOP policy is a dynamic policy that adjusts to the arrival of new information, the sequential approach is not likely to indicate the optimal route given the current state. The RSOP policy provides an expedient policy potentially available for use in our base optimization model. We report RSOP policy performance results to provide an additional benchmark policy to gauge ADP policy performance. We leverage Vansteenwegen & Gunawan (2019) to properly formulate the OP. We explicitly represent the mathematical formulation for the OP as

$$\max \quad \sum_{i=2}^{|N|-1} \sum_{j=2}^{|N|} P_i x_{ij}, \tag{19}$$

$$\text{s.t.} \quad \sum_{j=2}^{|N|} x_{1j} = \sum_{i=1}^{|N|-1} x_{i|N|} = 1, \tag{20}$$

$$\sum_{i=1}^{(|N|-1)} x_{ik} = \sum_{j=2}^{|N|} x_{kj} \leq 1; \ \forall \ k = 2, ..., (|N| - 1) \tag{21}$$

$$\sum_{i=1}^{(|N|-1)} \sum_{j=2}^{|N|} \rho_{ij} x_{ij} \leq \rho_t, \tag{22}$$

$$2 \leq u_i \leq |N|; \ \forall \ i = 2, ..., |N|, \tag{23}$$

$$u_i - u_j + 1 \leq (|N| - 1)(1 - x_{ij}); \ \forall \ i, j = 2, ..., |N| \tag{24}$$

$$x_{ij} \in \{0, 1\}, \forall \ i, j \in N \tag{25}$$

wherein we observe an objective to maximize the total reward by servicing nodes in the set $N = \{1, ..., |N|\}$. P_i denotes the reward received from visiting node i. The set of nodes also includes the current location of the current routed AUAV. The exit node (Ω) is the highest-indexed node in the set N. All other nodes in the system can be explicitly described as the set of targets in the attack domain at time t represented by $\{2, ..., (|N|-1)\} \subset N$. The system is constrained by Constraints (20) - (25) where Constraint (20) ensures each AUAV route starts from the starting node and ends at the departure node. Constraint (21) maintains the connectivity of each route and ensures each node is visited at most once. Constraint (31) ensures the limited time budget (ρ_t) for each route is not exceeded by summing the distance of each arc, denoted as ρ_{ij}. Constraint (23) and (24) prevent subtours from occurring in each route during the mission. We let u_i represent the position of node i in the AUAV path. Constraint (38) enforces the binary nature of the decision variables.Constraint (25) enforces the binary nature of the decision variables. Solving the OP to optimality determines the optimal routing for a single AUAV. After AUAV 1 declares its route, AUAV 2 then solves the OP for the remaining set of unvisited nodes and the optimal routing for AUAV 2 is declared.

The RSOP policy is a competitive routing policy for autonomous agents and largely exists to reduce the computational burden for use in our ADP algorithm, admittedly with the decomposition yielding a collectively greedy approach over AUAVs

that may yield a sub-optimal routing solution. However, implementing the RSOP policy still presents a large computational burden for use as the base optimization model; therefore, we develop the RGMH for use as the base optimization model in our ADP algorithm. The RGMH policy requires significantly less computational time while sacrificing minimal solution quality.

3.3.3 Repeated Greedy Marginal Heuristic Policy

The RGMH policy executes a greedy approach in collecting rewards based on the marginal value assigned to each target, where each AUAV is sequentially routed to a subset of target nodes in the attack domain. Our greedy marginal heuristic was motivated by Vansteenwegen *et al.* (2009a) and Pichpibul & Kawtummachai (2013) and is defined in Algorithm 1.

Algorithm 1 Greedy Marginal Heuristic

```
1. for m = 1 to M
2.     Determine AUAV m's location and playtime remaining
3.     while playtime > 0
4.       for n = 1 to N
5.         Calculate all targets available to AUAV m
6.         Calculate marginal value of target n
7.       end for
8.       Select target with maximum marginal value
9.       Adjust AUAVs position to selected target
10.      Adjust remaining playtime
11.    end while
12. end for
```

AUAV 1 selects its route based on all available targets given the state of the system. These targets are removed and AUAV 2 declares its route among the remaining targets in the attack domain. AUAV 1 begins by calculating all available targets to which it can travel and successfully return to the exit node given its allotted fuel resource. After this subset of targets is determined, the AUAV selects the one target that holds

the maximum marginal reward value, which is calculated via a function of the target's reward value and distance from the AUAV. The next target, n, added to the AUAV's path is selected by solving

$$\max_{n \forall N} \left\{ \frac{r_n}{d_n} \right\} \tag{26}$$

wherein r_n denotes the reward associated with servicing target n and d_n denotes the distance from target n to the AUAV. As the new target is added to the AUAVs route, the remaining playtime is decremented, and the AUAV changes its realized position to the current declared target location. The AUAV then repeats the process of determining the new subset of targets available during the remaining playtime, calculating all marginal values of targets, selecting the maximum marginal value, routing to that target, and adjusting the new set of targets available to the AUAV. When the set of remaining targets to visit is empty, the AUAV must route to the departure node. Once AUAV 1 finishes declaring the targets it visits, AUAV 2 looks at all remaining targets and proceeds through the same process. This policy produces routes for the team of AUAVs and generates competitive results to both the RTOP and RSOP policies with a large reduction in computational time. This primary advantage allows us to utilize the RGMH as the base optimization model in our ADP algorithm, where the policy is continually solved hundreds of thousands of times during our computational experiments.

3.4 ADP Solution Methodology

The well-known curses of dimensionality limit our ability to solve the MRP-DTA using traditional dynamic programming techniques. The curses of dimensionality refer to a high magnitude of dimensions that manifest in the state space, action space, outcome space, or any combination of the three. The curse of dimensionality hinders our ability to truly define the value of being in a certain state of the system (Powell,

2011). The MRP-DTA exhibits a prime example of a problem that suffers from the curse of dimensionality due to uncountable state and outcome spaces, requiring an ADP approach to derive a robust policy that exhibits anticipatory behavior.

Two fundamental categories of ADP solution techniques are notable in the field. These categories are lookahead approximations and policy search methods. Lookahead approximation techniques derive policies that produce the best current decision, given an estimation of the future impact of that decision. Lookahead approximation can be further subdivided into value function approximations (VFAs) and direct lookahead approximations (DLAs) and is a focus of the research community. A policy search method uses a parameterized model to derive a high-quality policy via gradient-based or gradient-free search techniques. This category of solution methods can be further subdivided into policy function approximations (PFAs) and cost function approximations (CFAs). We apply a CFA-DLA hybrid technique wherein we augment the cost function via a parameterization of problem features that helps to exploit the stochasticity in the system. Our base optimization model operates as a direct lookahead policy wherein the call to solve this model produces a projected performance for the team of AUAVs. Our hybrid approach provides improved results over the benchmark policies by parameterizing elements of the MRP-DTA that traditional optimization models do not include. Note that we refer to our deterministic optimization model used to solve the MRP-DTA as a cost function (although we are maximizing reward and not cost) to reflect the proper CFA terminology presented in Powell (2022).

3.4.1 Base Optimization Model

The embedded optimization model for the MRP-DTA is a deterministic model that can be solved, given any state of the system to produce a high-quality route for the team of agents. The MRP-DTA can be solved using mathematical programming techniques by formulating the problem as a team-orienteering problem (TOP). We are not able to directly solve the TOP formulation because solving for a solution to the TOP formulation requires a large amount of computational effort given the number of targets in the attack domain. We reference the base optimization model as our DLA component of our algorithm because it provides a deterministic forecast of the expected reward given the execution of the solution returned when the model is solved. This approach allows us to produce an approximation of the total reward for the team of AUAVs given any state of the system, classifying our model as a direct lookahead policy. Although the model remains the same, we evaluate propose several different policies for solving our deterministic model.

We proposed both a sequential routing policy that either solves the OP (RSOP policy) in Equations (19) - (25) or a greedy marginal heuristic (RGMH policy) via Equation (26). Each heuristic provides a possibly sub-optimal solution to the TOP mathematical formulation in a computationally efficient manner, which allows us to test and compare the use of each policy in solving the base optimization model in our ADP algorithm. We test the quality of solution obtained when utilizing the RSOP policy as our means for solving the base optimization model as well as when we use the RGMH policy as our means for solving base optimization model.

There are two primary means for parameterizing the embedded optimization model of the MRP-DTA: parameterization of the objective function and parameterization of the constraints. When a parameterization is applied to the objective function, we observe a soft bonus (or penalty) according to the set of parameterized

functions that are carefully crafted to exploit the stochasticity in the system of study. This approach assigns higher values to decisions according to a set of basis functions, which are deliberately crafted to handle the stochastic elements in the system. Furthermore, actions that may appear suboptimal in the original optimization model may be cost adjusted to appear more desirable after applying the soft bonus. We define the embedded optimization model with the applied soft bonus as follows

$$\max \quad \sum_{m=1}^{M} \sum_{i=2}^{|N|-1} P_i^{bonus} y_{im}, \tag{27}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} \sum_{j=2}^{|N|} x_{1jm} = \sum_{m=1}^{M} \sum_{i=1}^{|N|-1} x_{i|N|m} = M, \tag{28}$$

$$\sum_{m=1}^{M} y_{km} \leq 1; \ \forall \ k = 2, ..., (|N| - 1) \tag{29}$$

$$\sum_{i=1}^{|N|-1} x_{ikm} = \sum_{j=2}^{|N|} x_{kjm} = y_{km}; \ \forall \ k = 2, ..., (|N| - 1); \ \forall \ m = 1, ..., M \tag{30}$$

$$\sum_{i=1}^{(|N|-1)} \sum_{j=2}^{|N|} \rho_{ij} x_{ij} \leq \rho_t, \tag{31}$$

$$2 \leq u_{im} \leq |N|; \ \forall \ i = 2, ..., |N|; \ \forall \ m = 1, ..., M \tag{32}$$

$$u_{im} - u_{jm} + 1 \leq (|N| - 1)(1 - x_{ijm}); \ \forall \ i, j = 2, ..., |N|; \ \forall \ m = 1, ..., M \tag{33}$$

$$x_{ijm} \in \{0, 1\}, \forall \ i, j \in N; \ \forall \ m \in M \tag{34}$$

We denote P_i^{bonus} as the reward received for visiting target i after the soft bonus has been applied to target i. We augment the reward values in the objective function with no changes applied to the constraints of the problem which results in a pure parameterization of the objective function only. Note that the embedded optimization model is the same regardless of the means by which we choose to solve the model.

We analyze the means by which we solve the embedded optimization model (RSOP policy and the RGMH policy) in our results and analysis.

The soft bonus is applied to all targets, informed by the set of basis functions which help to represent the stochastic elements of the system. The successful construction of basis functions requires analysis of the MRP-DTA problem features and is modeled on the critical elements of the problem.

3.4.2 Basis Functions

The task of developing effective basis functions is essential to producing a high-quality routing policy for the MRP-DTA. Each basis function is a critical element in evaluating all decisions, given the state of the system. All soft bonuses applied to the set of reward values given a state of the system can be defined as

$$\sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t). \tag{35}$$

We represent f as a feature in the set of all features $\mathcal{F}$. Furthermore, $(\theta_f)_{f \in \mathcal{F}}$ is denoted as the linear weights or coefficients applied to the basis functions, $(\phi_f)_{f \in \mathcal{F}}$. This set of soft bonuses is then applied to the base optimization model to alter the decision-making process.

It is essential that we capture the most important features of the decision-making process to encourage optimal behavior. A high-quality dynamic routing policy is produced for the decision-maker by instantiating the proper basis functions. We focus our basis function construction on two important elements of the MRP-DTA that best exploit the stochasticity in the system: temporal and spatial basis functions. We parse the attack domain into a three-dimensional tile coding scheme to best value a target based on its position in the attack domain and the remaining playtime of the problem instance. A three-dimensional representation can be seen in Figure 6 where the x-

axis represents the x-coordinate of a target, the y-axis represents the y-coordinate of a target, and the z-axis represents the remaining playtime of the simulation. We

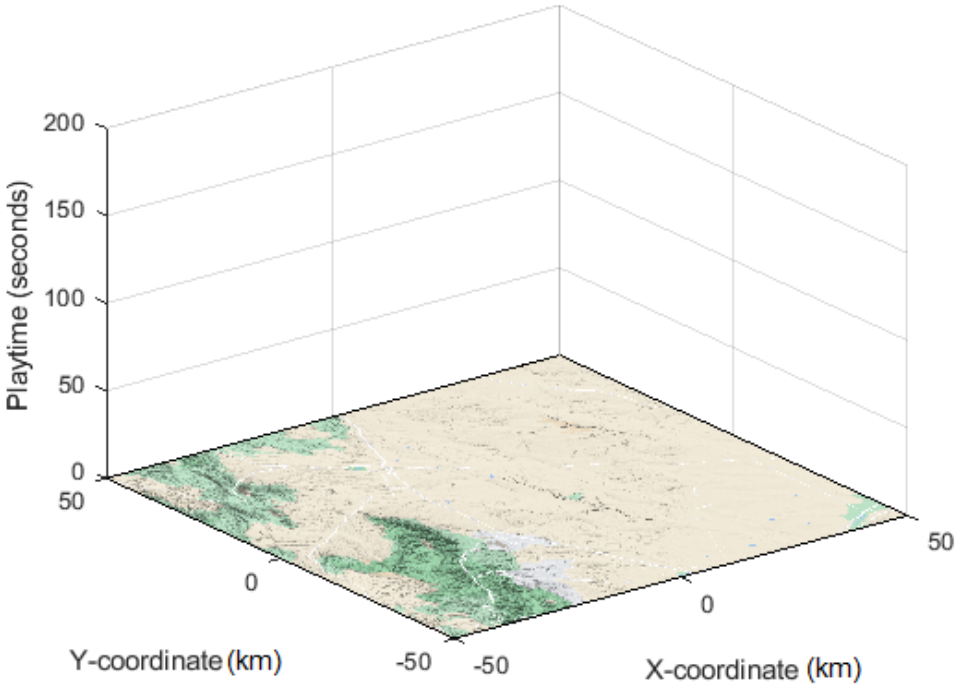

Figure 6. Three-Dimensional Space

specifically implement a tile coding scheme wherein we parse the target space into three-dimensions. The tile coding scheme aims to distinguish between targets that are spatially located throughout different regions of the attack domain and properly account for the remaining playtime for the team of AUAVs. A target that arrives early in the mission should indeed be preferred over a target that arrives late in the mission. Similarly, a target located in a reward-rich region (as defined by the probability of target arrivals) of the attack domain should be preferred over a target that arrives in a reward-poor region of the attack domain. We define L^w as the longitudinal width of the attack domain in kilometers and L^h as the latitudinal height of the attack domain in kilometers. We define the size of each x-axis discretization as Δ^x, which is calculated as the longitudinal width of the attack domain divided by the total number

of x-axis partitions. Similarly, we define the size of each y-axis discretization as Δ^y which is calculated as the latitudinal height of the attack domain divided by the total number of y-axis partitions. We define the width of each time interval as Δ^p, which is calculated as the total playtime of the simulation divided by the total number of playtime partitions. The set of basis functions that define the tile coding scheme can be written as

$$\phi_{nop}(S_t, x_t) = \begin{cases} 1, & \text{if } (n-1)\Delta^x - \frac{L^w}{2} < \ell^x_{tm} \leq (n)\Delta^x - \frac{L^w}{2}, \\ & \quad (o-1)\Delta^y - \frac{L^h}{2} < \ell^y_{tm} \leq (o)\Delta^y - \frac{L^h}{2}, \\ & \quad (p-1)\Delta^p < \rho_t \leq (p)\Delta^p \\ 0, & \text{otherwise.} \end{cases} \quad \forall \;\; n \in \mathcal{N}, o \in \mathcal{O}, p \in \mathcal{P}. \tag{36}$$

The target space is broken into $|\mathcal{N}||\mathcal{O}||\mathcal{P}|$ tiles to best capture the value of targets that fall within certain tiles of the three-dimensional grid, where $\mathcal{N}$ represents the set of x-axis partitions, $\mathcal{O}$ represents the set of y-axis partitions, and $\mathcal{P}$ represents the set of playtime partitions. We view each dimension's number of total partitions as a tunable parameter. It is generally true that a higher number of discretizations allows us to better exploit the system's structure at the cost of increasing computational effort (because we must tune a larger number of model parameters). We further investigate the proper discretization of the attack domain that produces high-quality routing policies for each instance of the MRP-DTA in our results and analysis.

To maximize total reward during the SCAR mission, the dispatching authority may be concerned with sending an AUAV to a remote location of the AOR (e.g., a part of the AOR with relatively few targets) due to the inherent disadvantage of losing large amounts of playtime. We develop two additional spatial basis functions that quantify a bonus associated with a target's relative location to the next closest target

and a bonus associated with a target's relative location to the geographic center of the attack domain. We define ψ_m^{clust} as the vector of distances that a target $m \in \mathcal{M}_t$ falls from all other targets in the attack domain and ψ_m^{center} as the distance that a target $m \in \mathcal{M}_t$ lies from the center of the attack domain. Utilizing this information, we develop the final two basis functions for use in our optimization model as

$$\phi_{clust}(S_t) = \min(\psi_m^{clust}) \quad \forall \quad m \in \mathcal{M}_t \tag{37}$$

$$\phi_{center}(S_t) = \psi_m^{center} \quad \forall \quad m \in \mathcal{M}_t. \tag{38}$$

Given all defined basis functions, we can produce a routing policy given a set of input parameters θ. Although a set of input parameters may be developed via real-time observation of the system and deliver high-quality results, this process would be classified as an engineering heuristic.

3.4.3 Mesh Adaptive Direct Search Algorithm

The derivative-free optimization method by which we tune the θ-parameters in our ADP optimization model is known as the mesh adaptive direct search (MADS) algorithm. The MADS algorithm is a derivative-free pattern search algorithm that minimizes a function W over the set of all possible input values, denoted as χ. Pattern search algorithms are beneficial when the gradient of W does not exist, or when it is difficult to estimate due to noise within W (Audet & Dennis Jr, 2006). The MADS algorithm builds upon a generalized pattern search (GPS) framework and provides an advantage over the GPS algorithm by not restricting the number of poll directions. This allows for an improved ability to locate an optimal objective function value.

The algorithm begins with an initial iterate denoted as $x_0 \in \chi$ and is executed in two fundamental steps. The search step begins at the beginning of each iteration,

k. In the search step, trial points are generated and their feasibility evaluated to determine whether they meet the constraints of the function W. If a trial point proves infeasible, the objective function value is set to a value that eliminates it from consideration as the current best-found solution. All other feasible trial points are then evaluated via the function W and then compared against what is known as the *incumbent point*. The incumbent point represents the current best-found solution and is denoted as x_k. We say that each trial point lies on the *mesh*, which is denoted as the finite set of directions scaled via a mesh size parameter, which we denote as $\Delta_k^m \in \mathbb{R}^+$. The objective of the search step is to find a point on the current mesh that beats the incumbent point. If a new incumbent point is found, the point replaces the current incumbent point, and the search step continues. This iterative process can be seen in Figure 7 from the transition from panel a to panel b. The search step terminates at

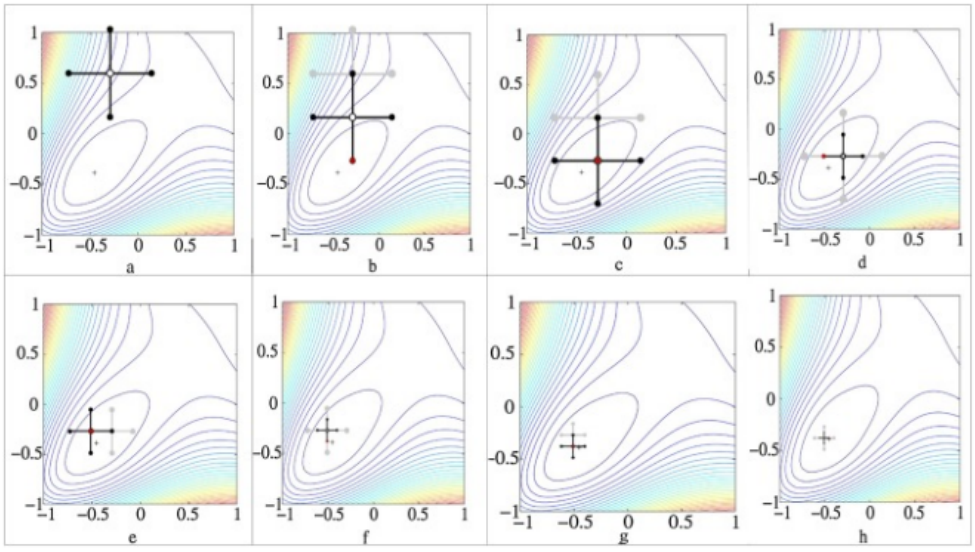

Figure 7. Pattern Search Procedure (Javed *et al.*, 2016)

the point at which a new incumbent point is not found among the mesh, and the poll step begins. The poll step initiates a local exploration around the current incumbent solution. The primary distinction between the MADS algorithm and GPS is found in the poll step, where we now consider an additional parameter known as the poll

size parameter, $\Delta_k^p \in \mathbb{R}^+$ (Audet & Dennis Jr, 2006). This parameter is a measure of magnitude that determines the distance from the current incumbent solution to the new trial points generated in the poll step. If the poll step fails to generate a new incumbent solution, the mesh size parameter and the poll size parameter are both reduced, and the next iteration begins. The increase in resolution can be seen in Figure 7 in the transition from panel c to d where the distance of trial points from the incumbent solution shrinks. A concise representation of this process can be seen in Figure 8. The MADS algorithm terminates whenever some stopping criterion is

A BASIC MADS ALGORITHM

Step 1: INITIALIZATION. Let x_k be such that $W(x_k)$ is finite. Let D be a positive spanning set, and let M_k be the initial mesh such that the initial mesh size parameter $\Delta_k^m \leq \Delta_k^p$ according to (9, 10).
Set the iteration counter $k \rightarrow 0$.

Step 2: SEARCH STEP. Evaluate W on a finite subset of trial points on the mesh M_k using some user defined strategy seeking an improved mesh point.

Step 3: POLL STEP. If the search step was unsuccessful or not performed, evaluate W on the frame P_k until an improved mesh point is found or all points in P_k have been evaluated.

Step 4: PARAMETER UPDATE. Update Δ_{k+1}^m according to (9), and Δ_{k+1}^p so that (10) is satisfied.
Set $k \leftarrow k+1$

Step 5: TERMINATION. If some stopping criterion is achieved, stop; if not, go back to step 2.

Figure 8. Basic MADS Algorithm (Hosseini *et al.*, 2011)

met. If in search of the global minimum value for the function f, then the stopping criterion is represented by a minimum mesh size threshold. The algorithm terminates whenever this threshold value is met, and the algorithm reports the current best-found solution.

We utilize the MADS algorithm to derive a competitive set of θ-parameters by

which we augment the embedded optimization model to produce a high-quality CFA policy. At the termination of the MADS algorithm, the best-found set of parameters is then returned which gives us our CFA policy denoted as π^{CFA}.

3.4.4 Algorithmic Strategy

The algorithmic strategy is the fundamental strategy by which we solve for a high-quality set of parameters, producing a high-quality routing policy for the MRP-DTA. We use a technique known as simulation optimization, where we simulate through sets of parameterizations (i.e., different policies) and evaluate their effectiveness when applied to the base optimization model. This requires the development of a system simulator that evaluates the effectiveness of a policy. The following simulation model seen in Figure 9 is the primary evaluation criteria by which we iteratively tune the θ-parameters. We do this by evaluating a complete sample trajectory, which is defined as the full execution of one instance of the MRP-DTA. By evaluating a sample trajectory and collecting a measure of total reward, we can obtain one sample point that describes the performance of the policy governed by the θ-parameters under evaluation. Due to the stochastic nature of the problem, a policy may perform well during one sample trajectory and poorly on the subsequent sample trajectory. This potential inconsistency presents a problem when trying to gauge true policy performance because the policy may appear to be very good over a small number of sample trajectories when its true policy performance is poor over a high number of sample trajectories. To best account for this issue, we simulate any given policy several times and take the average of the performance measure (i.e., total reward) across each replication to best estimate the policy's true solution quality.

At the simulator's core is a policy evaluation loop of size M. We test a set of θ-parameters by evaluating M different sample trajectories of the MRP-DTA and

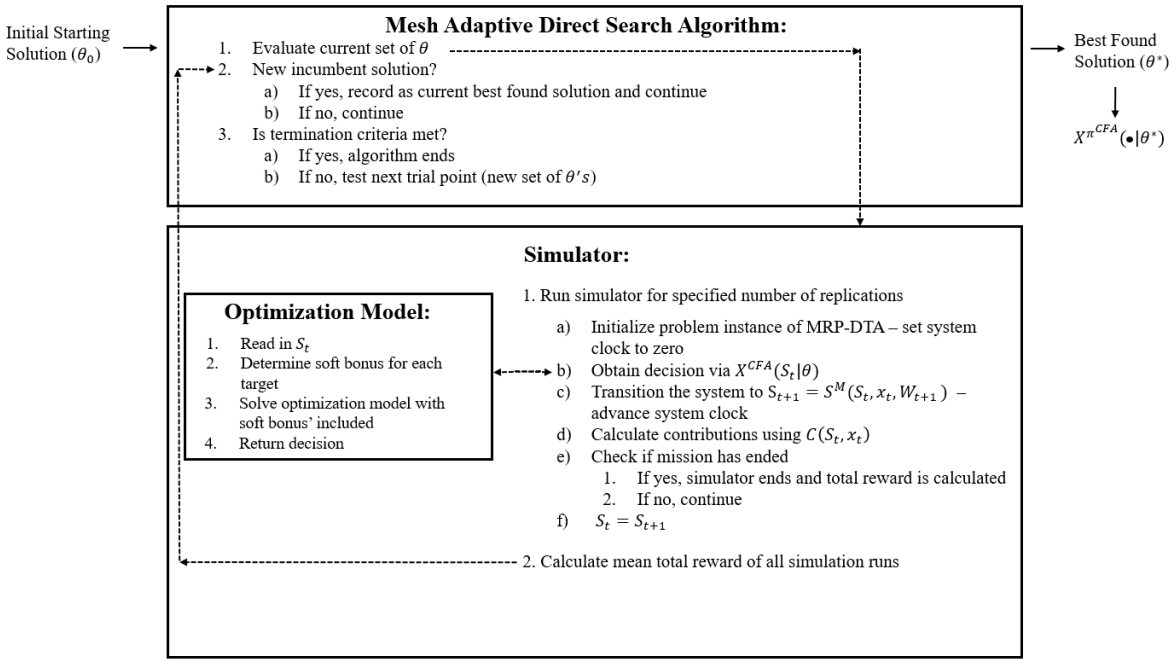

Figure 9. Graphical Depiction of Simulation Model

taking the mean total reward over all sample trajectories. Each individual run of the simulation requires a set of θ-parameters as input to the simulator. At the first step of the simulation, a fixed initialization of the MRP-DTA occurs according to the specified problem features. We generate a target arrival schedule that mimics an event schedule used in a discrete event simulation (Banks *et al.*, 2013). We use a target arrival schedule to concisely reference the times, locations, and types at which targets arrive in the system. Note that the arrival process and locations are still random; however, they are not generated in real-time but before the simulation's execution to simplify the process.

After initialization, we must obtain the routing decision for the team of AUAVs, which specifically calls our optimization model by passing in our set of θ-parameters. The optimization model determines the best decision according to the parameterization being applied to our base optimization model. The system is then transitioned to the next state based on the system model defined in Equation (3). At this moment,

we update the state variable and system clock (i.e., current playtime and the playtime remaining). The simulator then calculates any observed contributions based on the current state (S_t), the action taken (x_t), and the future state (S_{t+1}). The simulation then checks whether the terminal state has been reached. If not, we update the current state to the system's future state given by our system model in step 1c. When the simulation reaches the terminal state, we calculate the total reward obtained from the sample trajectory. At this point, another sample trajectory is calculated if we have not reached the specified number of simulation replications.

Whenever all simulation replications have occurred, we take the mean across all total rewards for each sample trajectory and report this as our performance metric for our set of θ-parameters. As we conduct a higher number of simulation replications, we can better predict the true solution quality for a set of θ-parameters. However, as we increase the number of simulation replications, we increase the computational effort. This introduces a fundamental tension in that we must balance the accuracy of our predicted solution quality with a reasonable amount of computational effort. If we extensively simulate a policy, we waste valuable computational time that may be used to evaluate other trial points. We treat this algorithmic parameter as a tunable parameter.

The MADS algorithm represents the policy improvement loop wherein we iterate through candidate solutions to find the best θ-parameters. The policy improvement loop requires the initialization of θ, which we chose to set to zeros. This particular initial solution represents the non-parameterized optimization model, which is equivalent to the benchmark policy result or the optimization model without any included soft bonus. We execute the policy improvement loop until a stopping criterion is met, namely a pre-established time limit, which is tunable by the user. At the moment the policy improvement loop terminates, the incumbent solution is returned as the best

found policy, denoted as $X^{\pi^{CFA}}(\cdot|\theta)$. This process is also described in the pseudo algorithm code in Algorithm 2.

Algorithm 2 (CFA-MADS Algorithm)

1. Initialize θ
2. **begin patternsearch(θ)**
3. **for** m = 1 to M
4. Initialize problem instance
5. $G_m = 0$
6. **while** $S_t \neq (\Omega, \Omega)$
7. Determine maximum action, $x_t = X^{CFA}(S_t|\theta)$
8. Simulate transition to next state, $S_{t+1} = S^M(S_t, x_t, W_{t+1})$
9. Record contribution, $G_m = G_m + C(S_t, x_t)$
10. **end while**
11. Take mean of Reward, $\bar{G} = \frac{1}{M}\sum G_m$
12. **end for**
13. Update θ via MADS algorithm
14. **end patternsearch()**
15. Return θ

IV. Testing, Analysis, and Results

In this chapter, we instantiate the multiagent routing problem with dynamic target arrivals (MRP-DTA) problem instance that serves as the basis for testing and analysis for our benchmark and ADP policies. This scenario serves as a standardized criterion for testing, analyzing, and discussing all policies. We design a multi-stage experiment to determine the critical problem features that drive the performance of our algorithms and to tune algorithm hyperparameters that maximize solution quality for our ADP algorithm. We simulate policy performance to gain insights into the behavior of each policy, leveraging common random numbers (CRNs) to reduce variance and improve computational efficiency. All computational experiments are conducted on an Intel Xeon Silver 4114 CPU, 2.20 GHz, 10-core processor with 64GB of RAM and MATLAB (2019a) parallel processing toolbox. We call IBM's CPLEX version 12.9.0 to solve all MILP formulations.

4.1 MRP-DTA Scenario

The MRP-DTA represents a team of two autonomous unmanned combat aerial vehicles (AUAVs) conducting a strike coordination and reconnaissance (SCAR) mission in an attack domain consisting of three unique target types. These target types are planned for destruction according to the joint force commander's (JFC's) objectives. The team of AUAVs acts as a forward observer (FO) for the AC-130U strike platform by providing lower altitude sensor capabilities that either mark targets for destruction by the AC-130U or reconnoiter named areas of interest (NAIs) established by the JFC. The team of AUAVs deploys from a forward arming and refueling point (FARP) with finite fuel resources and an initial route plan determined according to the joint integrated prioritized target list (JIPTL). As the mission commences, in-

telligence forces process dynamic target arrivals across the attack domain and add these to the JIPTL. Intelligence forces focus assets in regions of the attack domain known to have a higher probability density of target arrivals based on terrain features present in these regions. After fuel has been nearly exhausted, the AUAVs are routed back to the FARP to complete the mission.

The attack domain has two types of distinct terrain features that encompass each section of the attack domain: mountainous regions and desert regions. In application, these terrain types provide various benefits and pitfalls to the enemy. Each of the high-payoff targets (HPTs), regular payoff targets (RPTs), and NAIs are respectively assumed to hold the same inherent value and are assumed to be equally accessible to the team of AUAVs. In general, the JFC is responsible for aligning the target values considered by the team of AUAVs to their specific mission objectives. The notional JFC has approved of the fixed target values presented in Table 5.

Table 5. Approved JFC Target Values

Target Type	Reward Value
r^{HPT}	100
r^{RPT}	10
r^{NAI}	1

Recall these target values are crucial inputs to the contribution function defined in Equation (8). These target values motivate AUAV behavior in the attack domain. Although the AUAVs make decisions autonomously via algorithmic processing, they are managed by a dispatching authority that ensures the successful and lawful execution of the mission. As the mission commences, the dispatching authority has access to approve or deny a dynamic target arrival to the system if intermediary action is needed.

Intelligence forces process information on dynamic targets throughout the attack domain and build the JIPTL accordingly. Intelligence forces abide by the six-step

target processing cycle seen in Figure 5, wherein targets are scrutinized for potential efficacy. As previously mentioned, the process by which intelligence forces add dynamic targets to the JIPTL is a stochastic process that occurs according to a Poisson Process. The inter-arrival times between events are exponentially distributed according to an arrival rate λ. For example, an arrival rate of $\lambda = 1/20$ indicates an expected dynamic target arrival once every 20 minutes during the mission. It is desirable to assume the inter-arrival times between successive target arrivals are exponentially distributed because that probability distribution exhibits the memoryless property (Bain & Engelhardt, 1992). The memoryless property states that the future events in the system are independent of the occurrence of previous events in the system, which allows us to representatively model the dynamic target arrival process. Future research efforts may focus on the impact of this assumption on the MRP-DTA and further investigate the implementation of other stochastic processes such as the Hawkes process. The Hawkes process is a self-exciting point process wherein a point arrival increases the probability of an additional point arrival clustered nearby. Such an alternative assumption about dynamic target arrivals may affect the behavior of an AUAV and further reward anticipatory behavior.

Each AUAV is modeled after Boeing's X-45 joint unmanned combat aerial vehicle (J-UCAV) depicted in Figure 10. The X-45 is a joint DARPA, USAF, and United States Navy initiative developed primarily for strike missions (Air Force Technology, 2003).

The X-45 is outfitted with a Raytheon synthetic aperture radar, which provides a resolution of 60cm at a target range of 80km. The combat range of the X-45 is 2400 kilometers with a cruising speed of 0.8 mach. The MRP-DTA initializes both AUAVs with a maximum travel distance of 2400 kilometers at a speed of 0.8 mach (988 kilometers per hour) giving a total playtime of approximately 2.43 hours

Figure 10. Boeing X-45 J-UCAV

(145.7 minutes) (Air Force Technology, 2003). Recall that each AUAV travels at a uniform rate throughout its mission, expending a constant rate of fuel at all times. We disregard continuous curvature paths such as a Dubins path presented in Ismail *et al.* (2018); however, future research may formulate the MRP-DTA with this component to evaluate its impact on behavior.

We formulate the attack domain as 1,840 kilometers wide and 1,075 kilometers deep with the fire support coordination line (FSCL) and friendly troops positioned on the south border of the attack domain. The attack domain is specifically indexed via a military grid reference system (MGRS) with the orgin of the $\mathbb{R}^2$ plane denoting the very center of the attack domain. The FARP serves as both the starting node and the departure node for the team of AUAVs and is located on the southern border of the attack domain shown in Figure 11. Through intelligence preparation of the battlefield (IPB), intelligence forces have partitioned the attack domain into eight separate zones denoted by the markers in the top left corner of each rectangular region in the attack domain. These regions are distinguishable in terrain type and inform intelligence

assets during the mission. Intelligence forces focus assets in regions where targets are believed to be located. Initially, intelligence forces have declared that Regions 3, 4, 7, and 8 have a higher probability of target arrival of 20%. Regions 1, 2, 5, and 6 have a lower target arrival of only 5% due to the heavy focus in this area during the IPB phase of the targeting cycle. Intelligence forces have predicted a 40% chance that any incoming target is an HPT while the probability of a RPT target arrival is 60%. They establish a target arrival rate of $\lambda = 0.10$, which indicates an expectation of one target arrival every 10 minutes.

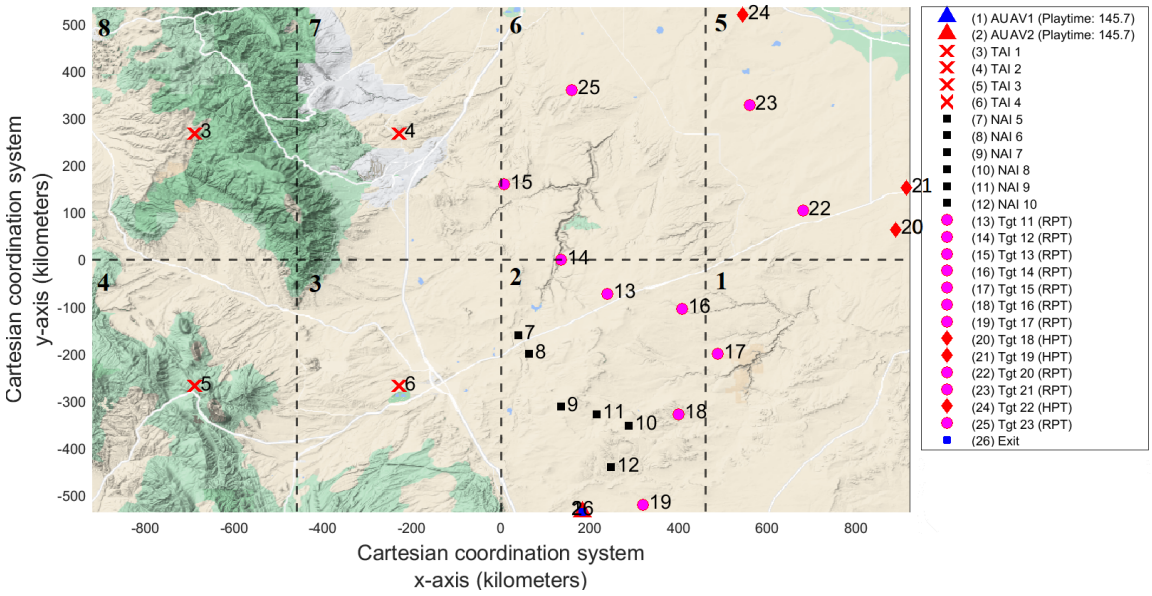

Figure 11. MRP-DTA Initialized Attack Domain in Matlab

The targeting process has produced an ATO for 19 approved deliberate targets in the attack domain. All target positions and target types can be see in Figure 11. We initialize the layout of targets in the attack domain according to a well investigated Orienteering Problem (OP) instance titled "Tsiligirides-set2-21" from Vansteenwegen & Gunawan (2019). The access to computational results for this problem set provides continuity to our research and helps to inform computational results. Figure 11 fully depicts the baseline starting scenario for the MRP-DTA.

4.1.1 Experimental Problem Features

The baseline representation of the MRP-DTA has many interesting problem features that drive system behavior and ultimately influence the execution of the mission. We use the term problem feature to describe any model parameter that defines the MRP-DTA setting. We wish to study how these problem features drive the development of our ADP policy and ultimately affect solution quality. These problem features give a good measure of policy robustness and help derive compelling problem instances.

A singular problem instance is represented by any unique set of problem feature settings. In the case of the MRP-DTA, multiple problem features exist, presenting a wealth of problem instances to test policy robustness. Specifically, we wish to design an experiment that selects the problem features most likely to indicate strong policy performance and explore the design space to find the settings at which any policy provides stellar results. Table 6 contains all experimental problem features for the MRP-DTA as well as the feature levels we choose to test.

Table 6. MRP-DTA Problem Features

Experimental Problem Features	**Feature Levels**
Target Arrival Rate	0.10**, 0.15
Probability of HPT	40%**, 80%
Prob Dist of Target Arrivals	Left**, Right, Top, Bottom, Split
Fixed Problem Features	**Fixed Feature Level**
AUAV Playtime (ρ_0)	145.75 minutes
Size of Attack Domain	1,840 km by 1,075 km
AUAV Speed	$16.46\bar{6}$ km/min
Number of NAIs	6 NAIs
Initial Number of Targets	19 HPTs, RPTs, NAIs
Initial Target Location	See Figure 11
Target Reward Values	See Table 5
Start/Exit Location	See Figure 11

* denotes the baseline problem setting

The choice to focus experimental resources on the three features in Table 6 results from these features tying closely to the stochastic nature of the system. The manifestation of stochasticity in these problem features provides the greatest insight to the dispatching authority, who is ultimately managing the behavior of each AUAV. By pinpointing the performance of both the benchmark policies and ADP policies at the exterior of the design space, we can provide a decision maker with depth of understanding. For both the target arrival rate and the probability of an HPT arrival, we choose to test at two different factor levels, which can be considered low and high factor levels. Our factor levels for the probability distribution of target arrivals across the attack domain exist as a categorical factor with five levels. We are interested in testing our ability to find a high-quality solution by altering the probability distribution of arrivals across the attack domain. Table 7 shows each categorical factor level's respective probability distribution over all eight regions in the attack domain.

Table 7. Probability Distribution for each Factor Level

	% Arrival Rate in Regions							
Factor Level Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Left	0.05	0.05	0.20	0.20	0.05	0.05	0.20	0.20
Right	0.20	0.20	0.05	0.05	0.20	0.20	0.05	0.05
Top	0.05	0.05	0.05	0.05	0.20	0.20	0.20	0.20
Bottom	0.20	0.20	0.20	0.20	0.05	0.05	0.05	0.05
Split	0.20	0.05	0.05	0.20	0.20	0.05	0.05	0.20

Next we generate our experimental design for the problem features of the MRP-DTA. We construct a full factorial design with three separate factors under evaluation. We are specifically examining at two factor levels for each continuous factor and five factor levels for our categorical factor. This decision results in a total number of 20 design points to test in our experiment. Testing various problem features helps to inform decision-making authorities on the sensitivity of each problem feature and can be informative to mission planning if certain problem features are uncertain

or undetermined. Mission planners can reference the sensitivity of each factor to determine "what-if" scenarios given uncertainty in problem features. Furthermore, by testing different problem features we are showcasing the robustness of our CFA-DLA algorithmic approach by demonstrating its ability to handle various problem instances of the MRP-DTA. The designed experiment for the problem features can be seen in Table 8.

Table 8. Experimental Design for Problem Features

Design Point	Point Designator	Arrival Rate	HPT Probability	Distribution of Arrivals
1	−−L	0.10	40%	Left
2	−−R	0.10	40%	Right
3	−−T	0.10	40%	Top
4	−−B	0.10	40%	Bottom
5	−−S	0.10	40%	Split
6	−+L	0.10	80%	Left
7	−+R	0.10	80%	Right
8	−+T	0.10	80%	Top
9	−+B	0.10	80%	Bottom
10	−+S	0.10	80%	Split
11	+−L	0.15	40%	Left
12	+−R	0.15	40%	Right
13	+−T	0.15	40%	Top
14	+−B	0.15	40%	Bottom
15	+−S	0.15	40%	Split
16	++L	0.15	80%	Left
17	++R	0.15	80%	Right
18	++T	0.15	80%	Top
19	++B	0.15	80%	Bottom
20	++S	0.15	80%	Split

4.1.2 Experimental Algorithm Parameters

Testing algorithm parameters is a common step in refining any ADP algorithm. Commonly referred to as tuning the algorithm, we are in search of the set of parameters producing the highest quality ADP policy given the MRP-DTA problem instance under study. It is common that an optimal algorithm setting for one problem instance does not resemble the optimal algorithm setting for a separate problem instance due to differences in stochastic elements of the system, which can often change the struc-

ture of the resulting policy.

For this reason, we design a computational experiment to tune each separate instance of the MRP-DTA to show the robustness of our algorithmic approach. We orchestrate our experimentation toward testing two separate algorithm parameters that drive the quality of solution. Table 9 contains all experimental algorithm parameters for the MRP-DTA as well as the algorithm parameters left constant.

Table 9. ADP Algorithm Parameters

Experimental Parameters	**Factor Levels**
Basis Function Selection	Tile Only, Tile-Cluster, Tile-Cluster-Center
Discretization of Tiling Scheme	$2 \times 2 \times 6$, $2 \times 2 \times 8$, $4 \times 4 \times 4$
Fixed Parameters	**Fixed Factor Level**
Total Runtime	12,000 secs (RGMH)/28,800 secs (RSOP)
Mesh Size Tolerance	1e-20
Initial Mesh Size	80 units
Inner Loop Size (M)	500 (RGMH)/100 (RSOP)

The choice to focus experimental resources on the two parameters in Table 9 results from extensive preliminary testing over a multitude of algorithm parameters. Preliminary experimentation is the process of briefly exploring parts of the design space in an attempt to save computational resources. This process can save tremendous resources in testing and analysis. We specifically found that the factor levels contained in Table 9 are sensitive factor levels that drive algorithm performance. These algorithm parameters directly affect the granularity of the tiling scheme, which is tied to the computational complexity of the algorithm. These parameters provide necessary discussion on the balance between granularity and computational effort. We evaluate each combination of algorithm parameters for each problem instance of the MRP-DTA to pinpoint the set of algorithm parameters that produce the highest solution quality. We generate the experimental design in Table 10, which defines the set factor levels for each design point.

Table 10. Experimental Design for Algorithm Parameters

Design Point	Point Designator	Tiling Scheme	Basis Function Selection
1	226-T	$2 \times 2 \times 6$	Tile Only
2	226-TC	$2 \times 2 \times 6$	Tile-Cluster
3	226-TCC	$2 \times 2 \times 6$	Tile-Cluster-Center
4	228-T	$2 \times 2 \times 8$	Tile Only
5	228-TC	$2 \times 2 \times 8$	Tile-Cluster
6	228-TCC	$2 \times 2 \times 8$	Tile-Cluster-Center
7	444-T	$4 \times 4 \times 4$	Tile Only
8	444-TC	$4 \times 4 \times 4$	Tile-Cluster
9	444-TCC	$4 \times 4 \times 4$	Tile-Cluster-Center

We construct a full factorial design with two separate factors under evaluation, specifically examining three factor levels for both factors. This results in a total number of nine design points by which we plan to test for each MRP-DTA problem instance.

We focus experimental resources on testing the granularity of our tiling scheme used in our basis function selection. The size of the tiling scheme is an important factor. The added granularity comes at a computational cost while increasing our ability to instantiate unique behavior. We also choose to test the selection of basis functions included in the algorithm. The addition of basis functions increases the dimensionality of our parameterization, thus increasing the computational cost of our algorithm. We evaluate the three separate basis function selections to determine the combination of basis functions that best balance solution quality with computational cost. Due to a limit on computational resources, we intentionally choose to investigate a Tile-Cluster scheme (and not a Tile-Center scheme) because preliminary testing showed a high performance for this combination. By tuning the algorithm parameters associated with each problem instance of the MRP-DTA, we demonstrate the robustness of our CFA-DLA algorithmic approach by showing its ability to derive a high-quality solution.

4.2 Experimental Results - RGMH Base Policy

The designed experiment conducted on the algorithm parameters for each problem feature setting yields the results seen in Table 11. Results are reported in terms of 95% confidence intervals for total reward (TR) over 1,000 different sample trajectories of the MRP-DTA. Although the inner loop size, M, is set to 500 sample trajectories, we simulate each policy over 1,000 sample trajectories to obtain a better measure of policy performance. The superlative algorithm setting (rows) are bolded for each problem feature setting (columns) and represents the setting with the highest mean TR when abiding by policy $\pi^{CFA-RGMH}$. We report the superlative policy from each problem instance and compare it to the benchmark policy performance. We report the mean and halfwidth for all policies. Recall Table 10 for information on the algorithm parameter settings for each design point and Table 8 for information on the problem instance settings for each problem instance.

The results show that selection of all basis functions does not provide superior results. When looking at all 20 problem instances under evaluation, the superlative policy utilizes all basis functions described in Section 3.4.2 in only 4 of the 20 instances. The inclusion of all basis functions increases the computational effort required to implement our algorithm while minimally improving the solution quality.

The $2 \times 2 \times 6$ tiling scheme produces the superlative algorithm settings in 9 of 20 problem instances. As we decrease the resolution of our tiling scheme, we decrease the computational effort that is required to solve our algorithm. If we apply a more granular approach, we increase the dimensionality of our parameterization which ultimately increases the computational requirements of our algorithm. The problem structure of the MRP-DTA requires an approach that balances the trade-off between granularity and computational cost. An interested party may choose to test a more granular approach if computational effort is not a concern to determine effectiveness

Table 11. DOE Results for Algorithm Parameters (CFA-RGMH)

Design Point	Prob Inst 1 (−−L)	Prob Inst 2 (−−R)	Prob Inst 3 (−−T)	Prob Inst 4 (−−B)	Prob Inst 5 (−−S)	Prob Inst 6 (−+L)	Prob Inst 7 (−+R)
1 (226-T)	475.17 ± 6.19	589.73 ± 7.85	512.43 ± 6.33	559.29 ± 7.79	$\mathbf{522.93 \pm 6.67}$	592.43 ± 8.99	764.87 ± 10.53
2 (226-TC)	480.75 ± 7.32	$\mathbf{595.60 \pm 7.98}$	511.19 ± 6.35	$\mathbf{564.10 \pm 7.69}$	520.17 ± 6.62	$\mathbf{629.11 \pm 9.26}$	$\mathbf{774.08 \pm 10.49}$
3 (226-TCC)	483.90 ± 7.38	591.15 ± 7.89	510.02 ± 6.36	560.14 ± 7.68	516.68 ± 6.67	628.93 ± 9.15	757.30 ± 10.35
4 (228-T)	474.61 ± 6.04	594.18 ± 7.98	$\mathbf{513.08 \pm 6.37}$	563.11 ± 7.61	517.81 ± 6.61	595.85 ± 9.16	771.26 ± 10.69
5 (228-TC)	474.33 ± 6.10	590.71 ± 7.91	510.78 ± 6.48	562.46 ± 7.86	519.49 ± 6.64	592.91 ± 9.05	767.63 ± 10.73
6 (228-TCC)	$\mathbf{485.63 \pm 7.33}$	588.41 ± 7.86	511.50 ± 6.43	562.13 ± 7.86	519.31 ± 6.63	627.96 ± 9.27	763.03 ± 10.48
7 (444-T)	477.22 ± 6.15	595.37 ± 7.91	512.04 ± 6.47	563.91 ± 7.77	517.41 ± 6.71	618.15 ± 9.35	772.81 ± 10.64
8 (444-TC)	479.96 ± 7.28	589.33 ± 7.82	511.84 ± 6.47	561.21 ± 7.75	517.66 ± 6.66	619.11 ± 9.54	759.86 ± 10.57
9 (444-TCC)	480.59 ± 7.20	588.85 ± 7.90	511.11 ± 6.47	559.79 ± 7.82	517.28 ± 6.71	619.97 ± 9.22	758.90 ± 10.78
Design Point	**Prob Inst 8 (−+T)**	**Prob Inst 9 (−+B)**	**Prob Inst 10 (−+S)**	**Prob Inst 11 (+−L)**	**Prob Inst 12 (+−R)**	**Prob Inst 13 (+−T)**	**Prob Inst 14 (+−B)**
1 (226-T)	629.99 ± 8.61	$\mathbf{755.37 \pm 11.33}$	$\mathbf{643.67 \pm 8.87}$	542.44 ± 8.39	681.88 ± 9.36	577.26 ± 7.63	653.43 ± 9.75
2 (226-TC)	629.16 ± 8.52	752.14 ± 11.27	642.68 ± 8.78	567.43 ± 8.80	688.12 ± 9.49	575.75 ± 7.69	664.66 ± 10.00
3 (226-TCC)	626.26 ± 8.82	754.83 ± 11.26	636.06 ± 8.83	567.41 ± 8.70	680.53 ± 9.30	575.65 ± 7.78	662.72 ± 10.09
4 (228-T)	629.71 ± 8.69	735.75 ± 11.20	639.16 ± 8.72	545.90 ± 8.32	$\mathbf{689.16 \pm 9.46}$	577.68 ± 7.79	659.24 ± 9.74
5 (228-TC)	631.09 ± 8.63	747.12 ± 11.43	637.01 ± 8.80	542.99 ± 8.21	683.62 ± 9.25	576.23 ± 7.55	653.82 ± 9.74
6 (228-TCC)	628.75 ± 8.72	755.24 ± 11.61	636.67 ± 8.72	$\mathbf{567.51 \pm 8.62}$	684.65 ± 9.46	574.05 ± 7.56	656.89 ± 9.86
7 (444-T)	$\mathbf{635.81 \pm 8.63}$	736.65 ± 11.47	642.42 ± 8.88	554.23 ± 8.66	680.88 ± 9.21	$\mathbf{578.76 \pm 7.86}$	659.79 ± 9.67
8 (444-TC)	631.15 ± 8.45	754.19 ± 11.13	634.88 ± 8.92	553.66 ± 8.84	681.28 ± 9.25	578.60 ± 7.65	$\mathbf{666.45 \pm 10.08}$
9 (444-TCC)	630.28 ± 8.75	754.68 ± 11.27	637.10 ± 8.87	564.23 ± 8.66	679.12 ± 9.46	575.63 ± 7.74	663.52 ± 10.03
Design Point	**Prob Inst 15 (+−S)**	**Prob Inst 16 (++L)**	**Prob Inst 17 (++R)**	**Prob Inst 18 (++T)**	**Prob Inst 19 (++B)**	**Prob Inst 20 (++S)**	
1 (226-T)	583.58 ± 7.89	723.44 ± 11.37	930.60 ± 12.14	740.38 ± 9.88	906.47 ± 13.50	747.86 ± 10.33	
2 (226-TC)	582.61 ± 7.92	753.12 ± 10.73	$\mathbf{941.23 \pm 12.10}$	741.27 ± 9.80	934.13 ± 13.05	742.91 ± 11.19	
3 (226-TCC)	582.10 ± 8.06	753.03 ± 10.71	921.79 ± 12.18	734.87 ± 9.81	$\mathbf{939.43 \pm 12.72}$	746.17 ± 11.39	
4 (228-T)	580.78 ± 7.86	720.71 ± 11.25	937.06 ± 12.06	743.92 ± 9.88	907.23 ± 13.71	745.10 ± 10.28	
5 (228-TC)	584.23 ± 7.96	723.80 ± 11.36	933.12 ± 12.04	738.66 ± 9.92	905.00 ± 13.29	746.76 ± 10.51	
6 (228-TCC)	585.25 ± 7.98	$\mathbf{757.49 \pm 10.79}$	922.02 ± 12.30	739.21 ± 9.91	930.88 ± 13.25	748.71 ± 11.04	
7 (444-T)	582.23 ± 7.99	726.54 ± 10.49	924.33 ± 12.43	738.31 ± 9.94	912.94 ± 13.40	$\mathbf{753.05 \pm 10.21}$	
8 (444-TC)	$\mathbf{587.54 \pm 7.96}$	740.53 ± 10.90	920.14 ± 12.17	$\mathbf{744.65 \pm 9.97}$	928.56 ± 12.89	751.17 ± 10.92	
9 (444-TCC)	581.39 ± 7.94	756.34 ± 10.78	916.48 ± 12.44	743.74 ± 9.98	937.19 ± 12.92	747.87 ± 10.72	

over a longer testing horizon. A more granular approach can help induce very specific behavior when certain situations are encountered, whereas a coarse tile coding scheme is more generalizable when applied to a large number of sample trajectories.

When a lower arrival rate is observed, the algorithm benefits from a coarse tiling scheme. The superlative policy is found in 7 of 10 instances (when $\lambda = 0.10$) when utilizing a $2 \times 2 \times 6$ tile coding scheme. When the arrival rate increases, we observe that the superlative algorithm setting is found in 8 of 10 runs when we apply a $2 \times 2 \times 8$ or $4 \times 4 \times 4$ tiling scheme. This trend is a result of the need for an increase in resolution as we introduce more target arrivals over the horizon. When target arrivals are more scarce, there is less benefit from a finer discretization of the attack domain; however, when we observe a high target arrival rate, the increase in resolution allows our framework to meticulously adjust the behavior of the AUAVs over the problem's time horizon. Our CFA-DLA framework is then able to tune the θ-values associated with the tiling scheme and induce high-quality behavior.

Next, we tabulate the results for all problem instances of study and denote the instances wherein our CFA-DLA approach provides major improvement over the RGMH benchmark policy. We focus our analysis on comparing our ADP algorithm against the benchmark policy to discuss the percentage improvement seen in overall TR. This metric is an unbiased result, allowing us to make fair comparisons across each problem instance. Results are displayed in Table 12. We perform a paired t-test and construct a 95% confidence interval around the difference in means. This allows us to declare a statistically significant difference between the benchmark and ADP policy (i.e., if the interval does not include 0).

The results show that $\pi^{CFA-RGMH}$ outperforms π^{RGMH} in 19 of 20 problem instances at a 95% confidence level. Problem Instance 13 fails to show a statistically significant difference between the benchmark and ADP policy, exhibiting a tie be-

Table 12. DOE Results for Problem Features (CFA-RGMH)

	π^{RGMH}		$\pi^{CFA-RGMH}$		$\pi^{CFA-RGMH}$ - π^{RGMH}			
Instance	**Mean**	**Halfwidth**	**Mean**	**Halfwidth**	**Mean**	**Halfwidth**	**% Improvement**	$\boldsymbol{\pi^{CFA-RGMH} \geq \pi^{RGMH}}$
1 (−−L)	473.75	6.10	485.63	7.33	11.88	5.20	2.51 %	623 of 1000
2 (−−R)	588.98	7.87	595.60	7.98	6.62	3.53	1.12 %	595 of 1000
3 (−−T)	510.46	6.45	513.08	6.37	2.62	1.57	0.51 %	874 of 1000
4 (−−B)	559.09	7.80	564.10	7.69	5.01	3.37	0.90 %	745 of 1000
5 (−−S)	517.36	6.73	522.93	6.67	5.57	2.52	1.08 %	762 of 1000
6 (−+L)	590.77	9.03	629.11	9.26	38.34	6.80	6.49 %	644 of 1000
7 (−+R)	756.65	10.42	774.08	10.49	17.44	5.54	2.30 %	632 of 1000
8 (−+T)	626.31	8.74	635.81	8.63	9.49	4.80	1.52 %	632 of 1000
9 (−+B)	725.88	11.30	755.37	11.33	29.49	7.36	4.06 %	586 of 1000
10 (−+S)	633.80	8.81	643.67	8.87	9.87	4.44	1.56 %	685 of 1000
11 (+−L)	540.53	8.20	567.51	8.62	26.98	5.91	4.99 %	666 of 1000
12 (+−R)	679.42	9.33	689.16	9.46	9.74	4.95	1.43 %	526 of 1000
13 (+−T)	575.10	7.75	578.76	7.86	3.67	4.19	0.64 %	530 of 1000
14 (+−B)	653.02	9.76	666.45	10.08	13.43	6.45	2.06 %	572 of 1000
15 (+−S)	580.73	7.93	587.54	7.96	6.81	2.84	1.17 %	731 of 1000
16 (++L)	715.73	11.36	757.49	10.79	41.77	7.48	5.84 %	692 of 1000
17 (++R)	917.16	12.18	941.23	12.10	24.07	7.52	2.62 %	613 of 1000
18 (++T)	735.11	9.85	744.65	9.97	9.54	6.09	1.30 %	637 of 1000
19 (++B)	904.11	13.43	939.43	12.72	35.32	9.22	3.91 %	613 of 1000
20 (++S)	740.70	10.18	753.05	10.21	12.35	7.09	1.67 %	571 of 1000

tween the two policies. Problem Instance 13 contains target arrivals in the top portion of the attack domain. When target arrivals are focused at the top of the attack domain, the benchmark policy is able to adjust to the target arrivals before entering the top regions. This advantage eliminates the benefit of anticipatory behavior using the developed parameterization and explains the low improvement over the benchmark policy. The highest improvements over the benchmark policy occur when target arrivals are focused in either the left or bottom portions of the attack domain. Problem Instance 16 denotes the highest improvement of 5.84% over the benchmark policy. When a high arrival rate and probability of an HPT arrival are observed in the left regions of the attack domain, the ADP policy is able to anticipate the high density of reward in the left regions, providing a significant improvement over the benchmark policy. We further analyze the total number of sample trajectories (out of 1,000) where our ADP policy outperforms the benchmark policy. Figure 12 shows a summary policy comparison between $\pi^{CFA-RGMH}$ and π^{RGMH} for all 1,000 sample trajectories.

Our ADP algorithm outperforms the benchmark policy in total number of sample trajectories for 13 out of 20 scenarios. These results inform a decision-maker on the stability of our stochastic system as well as the consistency of our superlative ADP algorithms. We have shown that the ADP policy is superior in solution quality to the benchmark for 19 of 20 problem instances; however, we see 7 problem instances where the benchmark outperforms the ADP policy in total number of sample trajectories, indicating that the non-anticipatory behavior induced by π^{RGMH} often succeeds. Despite this being the case, in 6 of these 7 instances, we have shown via our 95% confidence interval on the difference in means that our ADP policy is a higher quality team attack policy. This implies that, although the benchmark policy produces more consistent performance (for these problem instances), the ADP policy is expected to

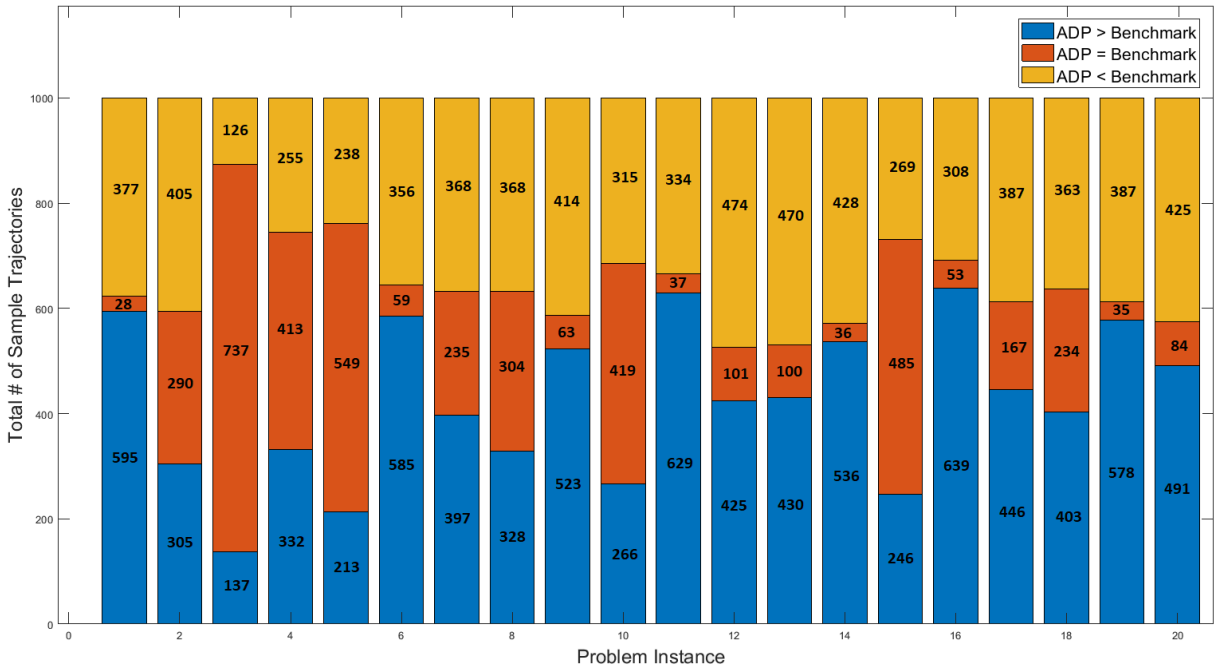

Figure 12. CFA-RGMH vs. RGMH Policy Performance Comparison over 1,000 Sample Trajectories of MRP-DTA

outperform the benchmark in solution quality over multiple executions of the problem instance.

We shift focus to the factors that impel a high percent improvement over the benchmark policy. Table 12 shows the percentage improvement of $\pi^{CFA-RGMH}$ over π^{RGMH}. To determine the factors in our problem instance experimental design that inform the response of percent improvement over the benchmark, we construct a second-order multiple linear statistical model that utilizes our factors from our experimental design to construct a metamodel that predicts the response of percentage improvement over the benchmark policy. Table 13 shows all parameter estimates for our model.

We construct a relatively strong model with an adjusted $R^2 = 0.8507$. This indicates that 85.07% of the variability in the response variable is accounted for in our regression model. When analyzing the parameter estimates in Table 13, the

Table 13. Parameter Estimates for Second-Order Linear Model

	Estimate	**Standard Error**	t **Ratio**	**Probability** $> \|t\|$
Intercept	2.47	0.13	18.70	$<$ **0.0001**
X_1	0.18	0.15	1.20	0.2627
X_2	0.74	0.15	5.00	**0.0011**
X_3 (Left)	2.59	0.27	9.74	$<$ **0.0001**
X_3 (Right)	-0.67	0.27	-2.51	**0.0361**
X_3 (Top)	-1.17	0.27	-4.42	**0.0022**
X_3 (Bottom)	0.39	0.27	1.47	0.1789
X_3 (Split)	-1.14	0.27	-4.28	**0.0027**
X_1X_2	-0.24	0.15	-1.60	0.1479
X_1X_3 (Left)	0.28	0.30	0.94	0.3761
X_1X_3 (Right)	-0.02	0.30	-0.07	0.9441
X_1X_3 (Top)	-0.20	0.30	-0.68	0.5168
X_1X_3 (Bottom)	0.07	0.30	0.25	0.8109
X_1X_3 (Split)	-0.13	0.30	-0.43	0.6757
X_2X_3 (Left)	0.46	0.30	1.56	0.1566
X_2X_3 (Right)	-0.15	0.30	-0.51	0.6262
X_2X_3 (Top)	-0.33	0.30	-1.10	0.3052
X_2X_3 (Bottom)	0.51	0.30	1.71	0.1248
X_2X_3 (Split)	-0.50	0.30	-1.68	0.1323
Curvature	0.22	0.17	1.33	0.2201
			Adj R^2	0.8507

* X_1 = Arrival Rate, X_2 = HPT Probability, X_3 = Distribution of Arrivals

factors that prove significant in predicting the response of percentage improvement over the benchmark policy are (1) probability of HPT arrival; (2) scenarios wherein target arrivals are focused in the left portion of the attack domain; (3) scenarios wherein target arrivals are focused in the right portion of the attack domain; (4) scenarios wherein target arrivals are focused in the top portion of the attack domain; (5) scenarios wherein target arrivals are split on the left and right portions of the attack domain. This result is indicated by the bolded p-values in the far right column of Table 13, and it informs future employment of the algorithm because the problem features that inform policy performance are the probability of a HPT target arrival and the location in which targets are expected to arrive. In an effort to further explore the design space and test for curvature in both our continuous factors (i.e., arrival rate and HPT probability), we augment our design with five additional center point runs to explore the design space at the central factor settings.

The concept of curvature in the design space is critical both to understand the influence of our continuous factors on the response and to generate insight regarding the performance of our ADP framework at the center point factor levels. We test arrival rate at $\lambda = 0.125$ and HPT probability at 60%. We perform one center point run for each categorical factor level of distribution of arrivals and use the resulting data to determine if curvature is present in our model. For each center point run, we tune our algorithm according to the experimental design shown in Table 10. The results obtained from the center point runs are contained in Table 14.

We observe that all five design points achieve a statistically significant improvement over the benchmark policy π^{RGMH}. We perform a two-sample t-test for curvature to determine whether curvature is present in the model. Our curvature term has a p-value of 0.22, thus indicating that curvature is not present.

Table 14. Center Point Run Results

	π^{RGMH}		$\pi^{CFA-RGMH}$		$\pi^{CFA-RGMH} - \pi^{RGMH}$		
	Mean	Halfwidth	Mean	Halfwidth	Mean	Halfwidth	% Improvement
1 (Left)	579.79	9.22	611.56	9.50	31.77	6.29	5.48 %
2 (Right)	744.14	10.29	755.69	10.21	11.54	5.12	1.55 %
3 (Top)	611.50	8.20	626.89	8.40	15.39	4.41	2.52 %
4 (Bottom)	712.73	10.82	736.90	11.00	24.17	7.08	3.39 %
5 (Split)	621.29	8.66	628.65	8.92	7.36	4.69	1.19 %

4.3 Experimental Results - RSOP Base Policy

We now apply the RSOP policy as the base optimization model and compare it against the RSOP benchmark policy. We intend to show that our ADP algorithmic framework can improve upon different base optimization models. Given the RSOP policy is more computationally expensive than the RGMH policy, we scale our experimentation accordingly by removing testing on the target arrival rate because this factor did not exhibit a statistically significant effect when utilizing the RGMH base optimization model. We leave the target arrival rate constant at $\lambda = 0.10$ to reduce our problem instance DOE from 20 to 10 design points.

The designed experiment conducted on the algorithm parameters for each problem feature setting yields the results seen in Table 15. Results are reported in terms of mean total reward (TR) executed over 100 different sample trajectories of the MRP-DTA where the bolded values represent the superlative policy for each problem instance. Note that we intentionally scale the number of sample trajectories from 500 to 100 total evaluations ($M = 100$) in an effort to decrease the computational cost of evaluating a candidate policy within our policy evaluation loop. When executing a lower number of sample trajectories in our policy evaluation loop, we run the risk of returning a policy that overfits to the low number of evaluated sample trajectories. To mitigate this risk, we simulate each bolded policy over 500 sample trajectories, construct a 95% confidence interval, and compare against π^{RSOP}. The superlative al-

gorithm settings (rows) are bolded for each problem instance (columns). We observe improved policy performance (over 500 sample trajectories) when evaluating Design Point 5 for Problem Instance 7 instead of Design Point 8. Design Point 8 for Problem Instance 7 was noted as performing poorly when simulated over 500 sample trajectories whereas we found Design Point 5 for Problem Instance 7 performs exceptionally well. As with previous testing, Tables 8 and 10 respectively indicate the problem instance settings and algorithm parameter settings for each design point.

The results show that the inclusion of the cluster basis function is critical to enhancing algorithm performance. The superlative policy is found in 9 of 10 instances when we leverage the implementation of the tiling scheme and the cluster basis function. Furthermore, the results indicate that utilizing a $2 \times 2 \times 8$ tiling scheme is superior to the other tiling schemes under evaluation. This emphasizes the temporal characteristics of a target arrival. It can be advantageous to alter the team behavior with respect to the remaining playtime when a target arrives so as to better position AUAVs in the attack domain.

The $2 \times 2 \times 6$ tiling scheme produces the superlative policy when target arrivals are focused in the left portion of the attack domain. This tiling scheme significantly decreases the computational effort required to solve our algorithm. If we apply a more granular approach, we increase the dimensionality of our parameterization, which ultimately increases the computational requirements of our algorithm. When target arrivals are focused in the left regions of the attack domain, our algorithmic framework requires less granularity likely because the need to alter AUAV behavior over the horizon of the sample trajectory is irrelevant. An interested party may choose to test a more granular approach if computational effort is not a concern to determine effectiveness.

Next, we tabulate the results for all problem instances of study and denote the

Table 15. DOE Results for Algorithm Parameters (CFA-RSOP)

Design Point	Prob Inst 1 (−−L)	Prob Inst 2 (−−R)	Prob Inst 3 (−−T)	Prob Inst 4 (−−B)	Prob Inst 5 (−−S)
1 (226-T)	484.19	615.02	527.12	579.07	524.55
2 (226-TC)	**501.51**	623.87	527.28	589.49	523.99
3 (226-TCC)	484.29	622.08	527.12	579.17	523.81
4 (228-T)	484.43	612.14	528.23	578.56	519.11
5 (228-TC)	488.44	**633.75**	**535.20**	583.70	**539.08**
6 (228-TCC)	500.22	618.78	527.44	**589.96**	519.11
7 (444-T)	485.99	612.57	527.22	582.16	526.93
8 (444-TC)	487.79	622.24	528.55	581.63	523.76
9 (444-TCC)	489.79	619.96	527.65	579.82	527.70
Design Point	**Prob Inst 6 (−+L)**	**Prob Inst 7 (−+R)**	**Prob Inst 8 (−+T)**	**Prob Inst 9 (−+B)**	**Prob Inst 10 (−+S)**
1 (226-T)	610.87	808.48	657.79	730.80	660.28
2 (226-TC)	**626.29**	807.58	664.68	734.59	655.09
3 (226-TCC)	611.80	808.95	654.14	726.13	652.01
4 (228-T)	610.76	808.64	654.89	727.48	652.12
5 (228-TC)	610.07	**808.14**	**667.41**	732.54	**663.17**
6 (228-TCC)	614.85	809.37	653.75	730.00	651.65
7 (444-T)	613.87	809.13	654.32	733.69	652.18
8 (444-TC)	611.11	809.59	654.71	**755.71**	652.36
9 (444-TCC)	614.71	802.23	653.72	729.14	653.13

instances where our CFA-DLA ADP approach provides major improvement over the RSOP benchmark policy. We focus our analysis on comparing the benchmark policy against our ADP algorithm and discuss the percentage improvement seen in overall TR. This metric is a unbiased result, allowing us to make fair comparisons across each problem instance tested. Results are displayed in Table 16. We perform a paired t-test and construct a 95% confidence interval around the difference in means. This allows us to declare a statistically significant difference between the benchmark and ADP policy (i.e., if the interval does not include 0).

The results show that $\pi^{CFA-RSOP}$ outperforms π^{RSOP} in 8 of 10 problem instances. Problem Instance 3 and Problem Instance 8 fail to show a statistically significant difference between the benchmark and ADP policy, thus exhibiting a tie between the two policies. When target arrivals are focused at the top of the attack domain, the benchmark policy is able to adjust to the target arrivals before entering the top regions. This advantage eliminates the benefit of anticipatory behavior using the developed parameterization. We observe that the highest improvements over the benchmark policy occur when target arrivals are focused in the left portions of the attack domain. This result is due to the inherent benefit of tasking one AUAV to

Table 16. DOE Results for Problem Features (RSOP)

	π^{RSOP}		$\pi^{CFA-RSOP}$		$\pi^{CFA-RSOP}$ - π^{RSOP}			
Instance	**Mean**	**Halfwidth**	**Mean**	**Halfwidth**	**Mean**	**Halfwidth**	**% Improvement**	$\boldsymbol{\pi^{CFA-RSOP} \geq \pi^{RSOP}}$
1 (−−L)	494.63	8.73	504.97	8.92	10.35	5.53	2.09 %	351 of 500
2 (−−R)	617.39	11.04	627.75	11.36	10.36	5.79	1.68 %	316 of 500
3 (−−T)	532.12	9.16	532.78	8.98	0.67	4.30	0.13 %	331 of 500
4 (−−B)	580.82	10.57	587.49	11.20	6.67	6.23	1.15 %	305 of 500
5 (−−S)	531.70	9.02	538.44	9.54	6.74	4.71	1.27 %	300 of 500
6 (−+L)	611.55	12.43	623.39	11.82	11.84	8.26	1.94 %	322 of 500
7 (−+R)	800.03	15.37	808.25	14.83	8.22	7.67	1.03 %	301 of 500
8 (−+T)	654.91	12.40	659.08	12.38	4.17	6.48	0.64 %	307 of 500
9 (−+B)	743.86	15.34	756.75	16.08	12.89	9.51	1.73 %	281 of 500
10 (−+S)	655.72	12.52	666.63	12.63	10.91	6.65	1.66 %	308 of 500

service dynamic targets while the other AUAV services deliberate targets. We further analyze the total number of sample trajectories (out of 500) where our ADP policy outperforms the benchmark policy. Figure 13 shows a complete policy comparison between $\pi^{CFA-RSOP}$ and π^{RSOP} for all 500 simulated sample trajectories.

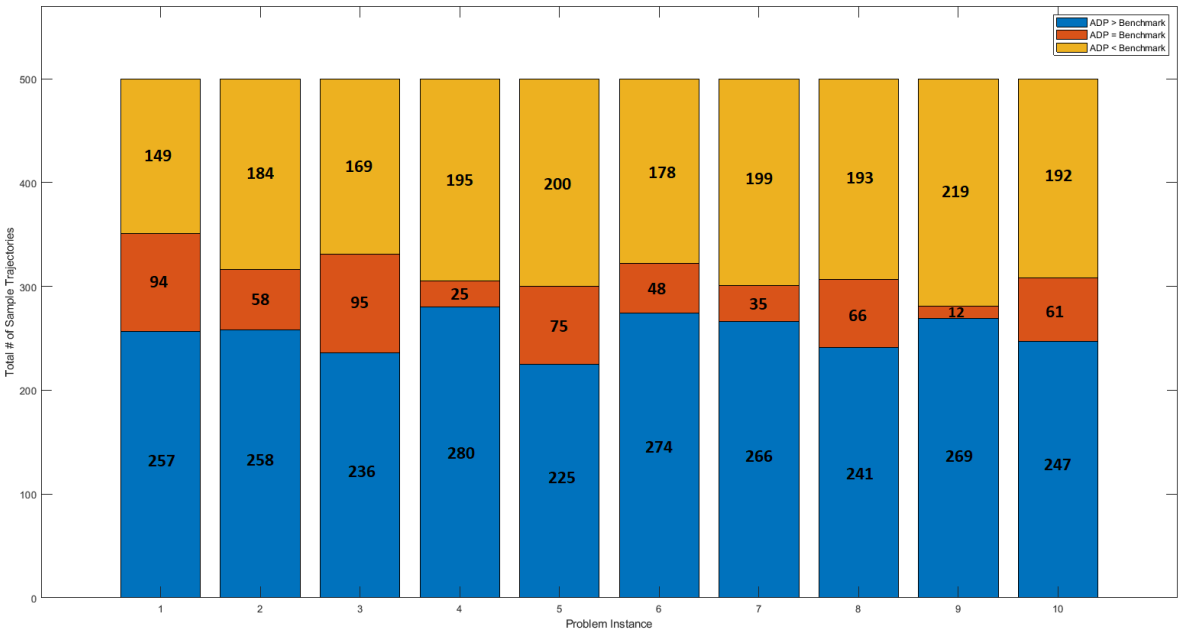

Figure 13. CFA-RSOP vs. RSOP Policy Performance Comparison over 500 Sample Trajectories of MRP-DTA

Our ADP algorithm outperforms the benchmark policy in total number of sample trajectories for 10 out of 10 scenarios. These results indicate that $\pi^{CFA-RSOP}$ is a consistent policy that performs well against the benchmark policy. This element is critical to the implementation of $\pi^{CFA-RSOP}$ because stakeholders desire a policy that provides consistent performance improvements. The results show that our ADP policies perform as good as or better than the benchmark policies in the majority of sample trajectories for all problem instances of study.

4.4 Case Study Evaluation

We perform a case study evaluation of problem instances to explain the superior performance of our ADP algorithm when compared against the respective benchmark policy. Case study evaluation provides stakeholders with meaningful insight into model algorithm performance and the aspects of the MRP-DTA that are crucial to the decision-making process. We specifically wish to identify instances wherein anticipatory behavior contributes to superior ADP policy performance and other behavioral characteristics that provide an improvement in TR for the team of AUAVs. We explicitly highlight two instances wherein the augmentation of the base optimization model produces a high-quality anticipatory policy.

Problem Instance 16 of our RGMH policy DOE exhibits a primary example of superior ADP policy performance. For Problem Instance 16, target arrivals are focused in the left portion of the attack domain. We observe Sample Trajectory 291, which showcases a trajectory in which $\pi^{CFA-RGMH}$ performs exceptionally well due to anticipatory behavior. First we consider the benchmark policy performance. In the top frame of Figure 14, both AUAVs are routed to the right portions of the attack domain due to the presence of high payoff deliberate targets.

As previously discussed, the benchmark policy is a reactionary policy. The AUAVs can adjust to target arrivals; however, they do not utilize the TAIs in anticipation of the target arrivals. This reactionary behavior causes the red AUAV to waste valuable playtime early in the sample trajectory by traveling toward the top of the attack domain. The middle frame of Figure 14 exhibits a prime example of reactionary behavior. Node 26 arrives to the attack domain and the red AUAV adds this target to the JIPTL and begins to move toward this target. Although the red AUAV adjusts to the dense target arrivals in the left portion of the attack domain later in the sample trajectory, the red AUAV has wasted valuable playtime by failing to anticipate these

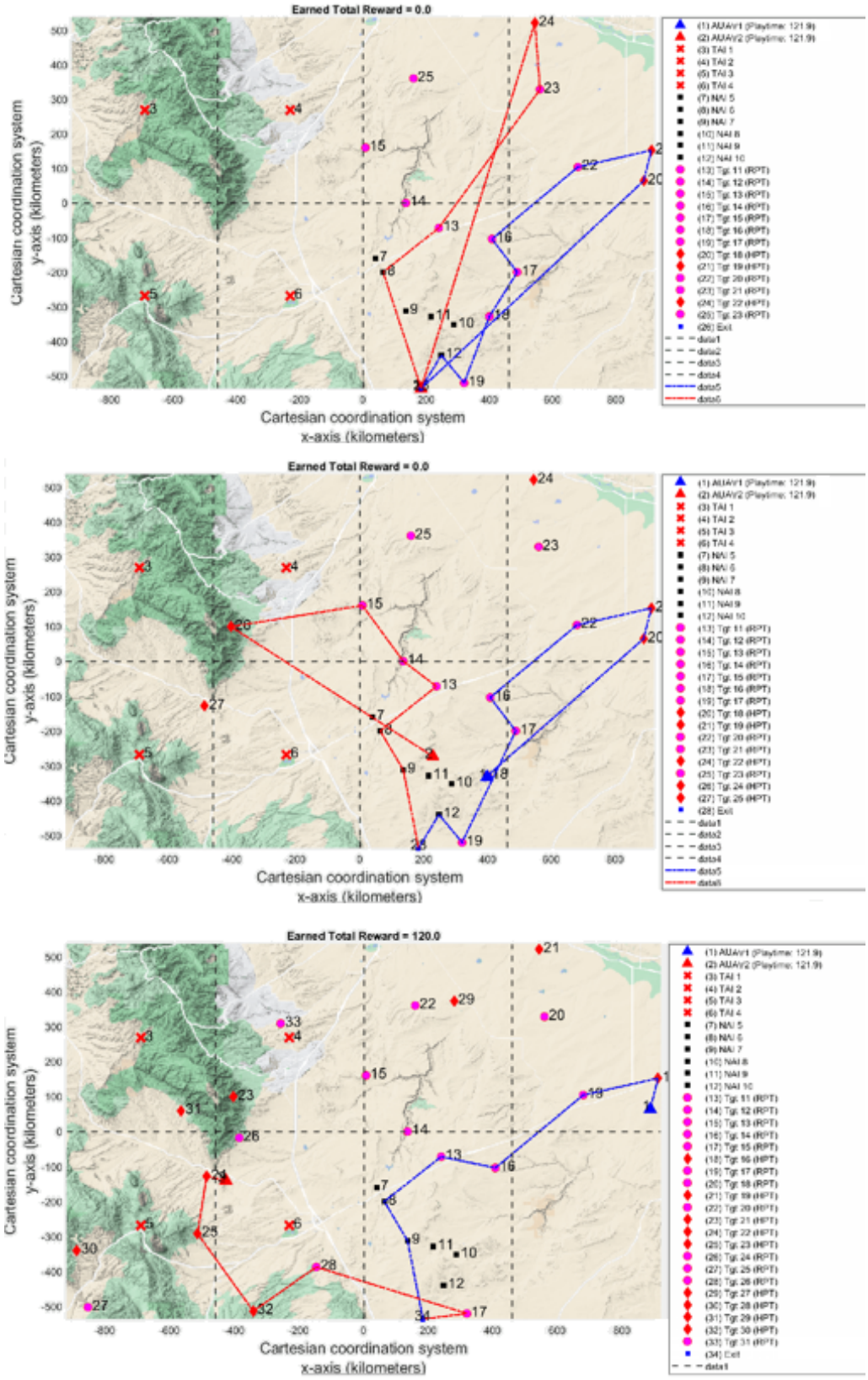

Figure 14. Sample Trajectory 291: π^{RGMH} (Benchmark) Policy

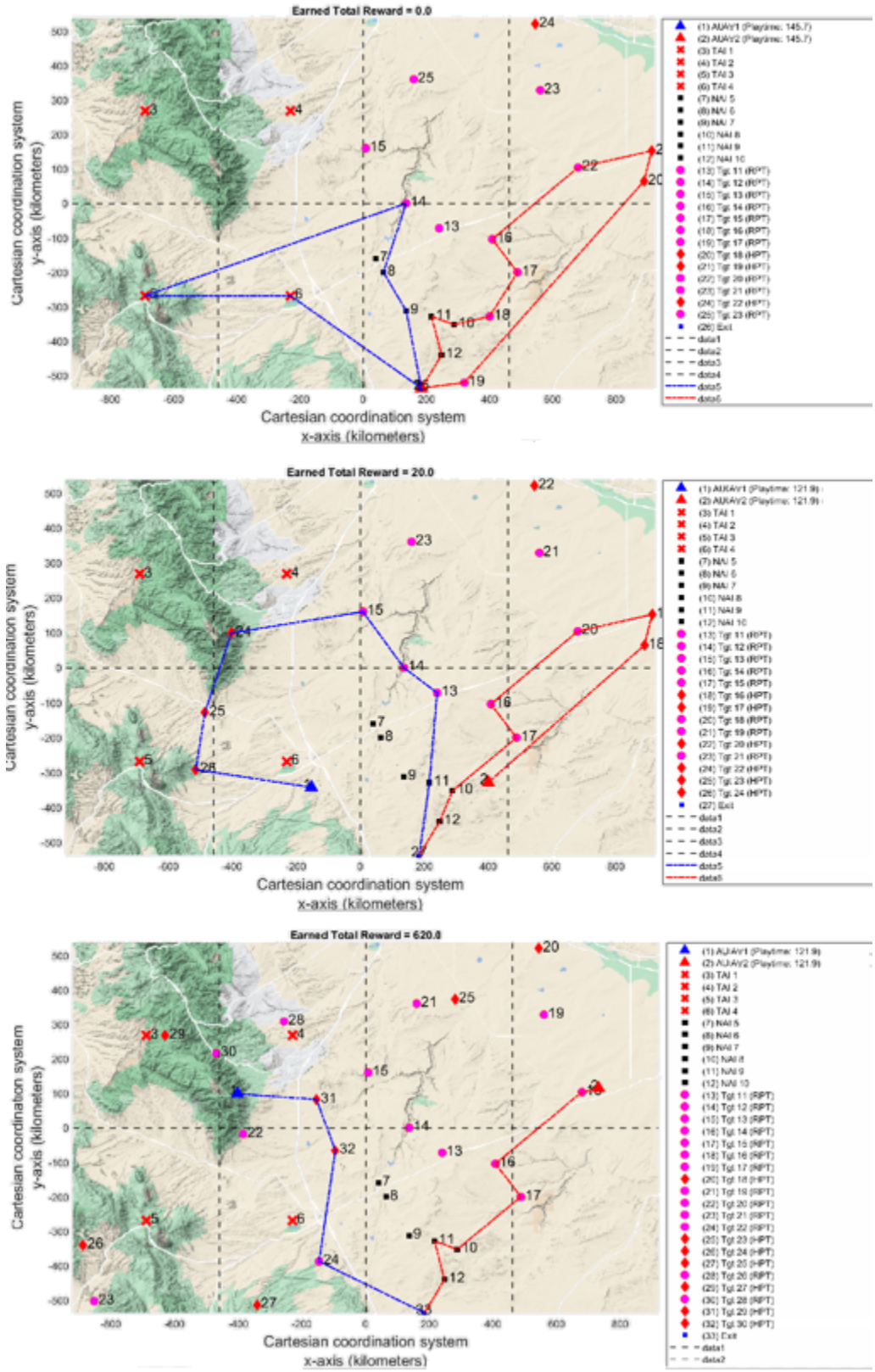

Figure 15. Sample Trajectory 291: $\pi^{CFA-RGMH}$ (ADP) Policy

target arrivals. The red AUAV heavily benefits from this route adjustment. The bottom frame of Figure 14 shows where the red AUAV is able to add three HPTs to its remaining, revised route. Recall the initial routing of the red AUAV, which included only one HPT. In total, the benchmark policy yields a total reward of 581 points. In Figure 15, we observe the same sample trajectory of the MRP-DTA except the AUAVs now abide by policy $\pi^{CFA-RGMH}$. The ADP policy $\pi^{CFA-RGMH}$ learned to anticipate target arrivals in the left regions of the attack domain by choosing to initially route the blue AUAV to TAI 4, which saves time for the blue AUAV later in the sample trajectory. Ultimately, the blue AUAV moves into the left region of the attack domain because it learned that a high density of target arrivals occur in the left subregions. Due to the time saved by the blue AUAV, it is able to service an additional 5 HPTs in the left central region of the attack domain. In total, the ADP policy yields a total reward of 870 points, which is a total improvement 289 points over the benchmark policy.

This sample trajectory highlights the fundamental advantage of our ADP policy. The ADP policy learns via the MADS algorithm, a high quality parameterization to apply to the base optimization model. The parameterization of the objective function in the base optimization model is the means by which we alter the behavior of the team of AUAVs and induce anticipatory behavior. We display the θ-values in Figure 16 to further analyze the parameterization that is being applied to the base optimization model. We observe a θ_1-value of 80 units of reward applied to any target that arrives in between the playtime bounds of 127.53 minutes and 145.75 minutes. This augmentation applies to deliberate targets since they are assumed to have arrived with 145.75 minutes of playtime remaining. The soft bonus applies to both Node 5 and Node 6, which causes the blue AUAV to route to these TAIs early in the sample trajectory. No value is applied to either the cluster basis function or

the centroid basis function, resulting in a pure tile coding scheme being applied to the model. As playtime is reduced throughout the simulated trajectory, we observe negative values applied to the θ_1 and θ_2 regions of the attack domain.

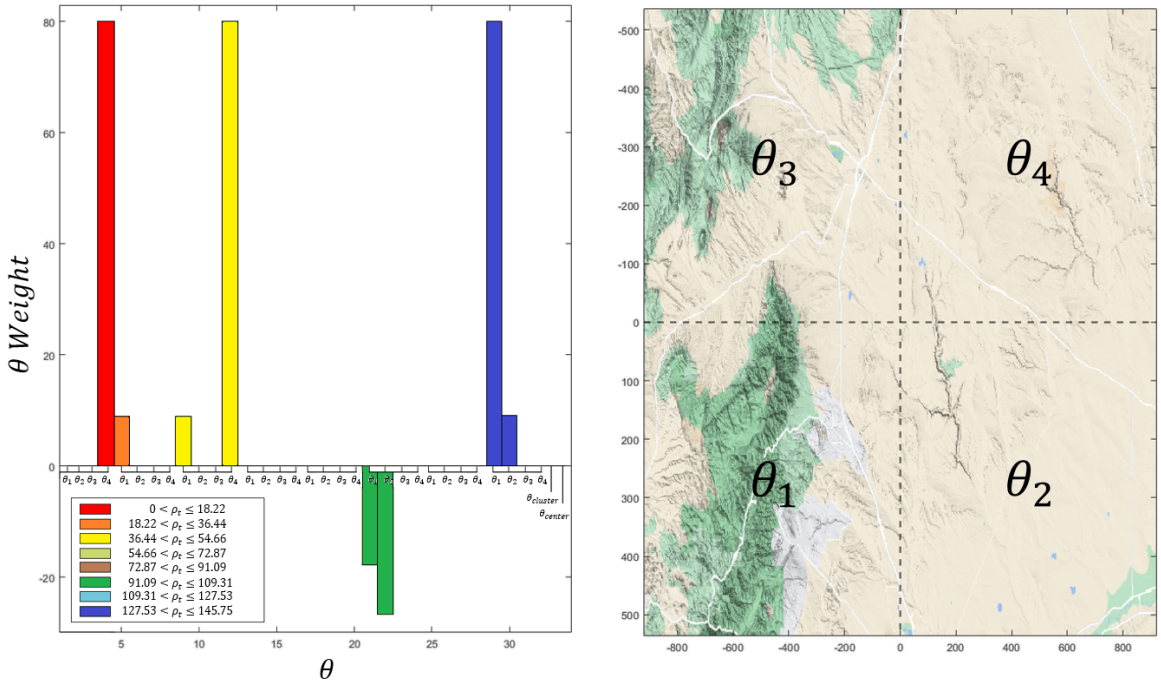

Figure 16. θ-values for Parameterization of ADP Policy for Problem Instance 16

The θ-value parameterization induces each AUAV to move toward the upper quadrants of the attack domain to gain an inherent positional advantage. When the AUAVs move to the upper quadrants of the attack domain, they are provided with a superior position to reroute and attack targets in the bottom quadrants during their movements back to the departure node. This behavior is exhibited by the adjustments seen to the blue AUAV's route when comparing the middle frame to the bottom frame in Figure 15. A heavy parameterization is then applied to θ_4 in the $36.44 \leq \rho_t \leq 54.66$ playtime tile to further encourage the AUAVs to visit the upper quadrant of the attack domain. These parameterizations are the primary means by which our ADP policy achieves anticipatory behavior and superlative results over π^{RGMH}. Our CFA-DLA framework is able to learn high-quality behavior exhibited in

the above parameterization given the problem features applied to Problem Instance 16.

Problem Instance 9 of our RGMH policy experimental design exhibits an additional example of superior policy performance of the ADP algorithm. For Problem Instance 9, target arrivals are focused in the bottom portion of the attack domain. We highlight Sample Trajectory 698, which showcases another trajectory wherein $\pi^{CFA-RGMH}$ performs exceptionally well due to anticipatory behavior. First, we discuss the performance of the benchmark policy. The top frame of Figure 17 shows both AUAV's declared routing for the problem instance. Recall that the benchmark policy is able to adjust to target arrivals after they occur but will not show anticipatory behavior.

In the studied instance, neither AUAV chooses to markedly adjust its route throughout the sample trajectory when abiding by π^{RGMH}. Each AUAV slightly adjusts to target arrivals that occur in the right portion of the attack domain. The middle frame of Figure 17 exhibits an example in which the red AUAV adds Node 28 to the JIPTL for scheduled destruction. Although the team of AUAVs achieves a TR of 672 throughout the sample trajectory, the bottom frame of Figure 17 shows a dense arrival of targets that occur in the bottom left portion of the attack domain that are unvisited throughout the sample trajectory.

In Figure 18, we observe the same sample trajectory of the MRP-DTA where the AUAVs now abide by policy $\pi^{CFA-RGMH}$. The ADP policy $\pi^{CFA-RGMH}$ anticipates target arrivals in the left regions of the attack domain by choosing to initially route the blue AUAV to Node 6. Ultimately, the blue AUAV moves into the left region of the attack domain because it learns that a high density of target arrivals occur in Region 4 of the attack domain. The blue AUAV is able to include these target arrivals and improve on TR for the team of AUAVs by achieving a TR of 870 points.

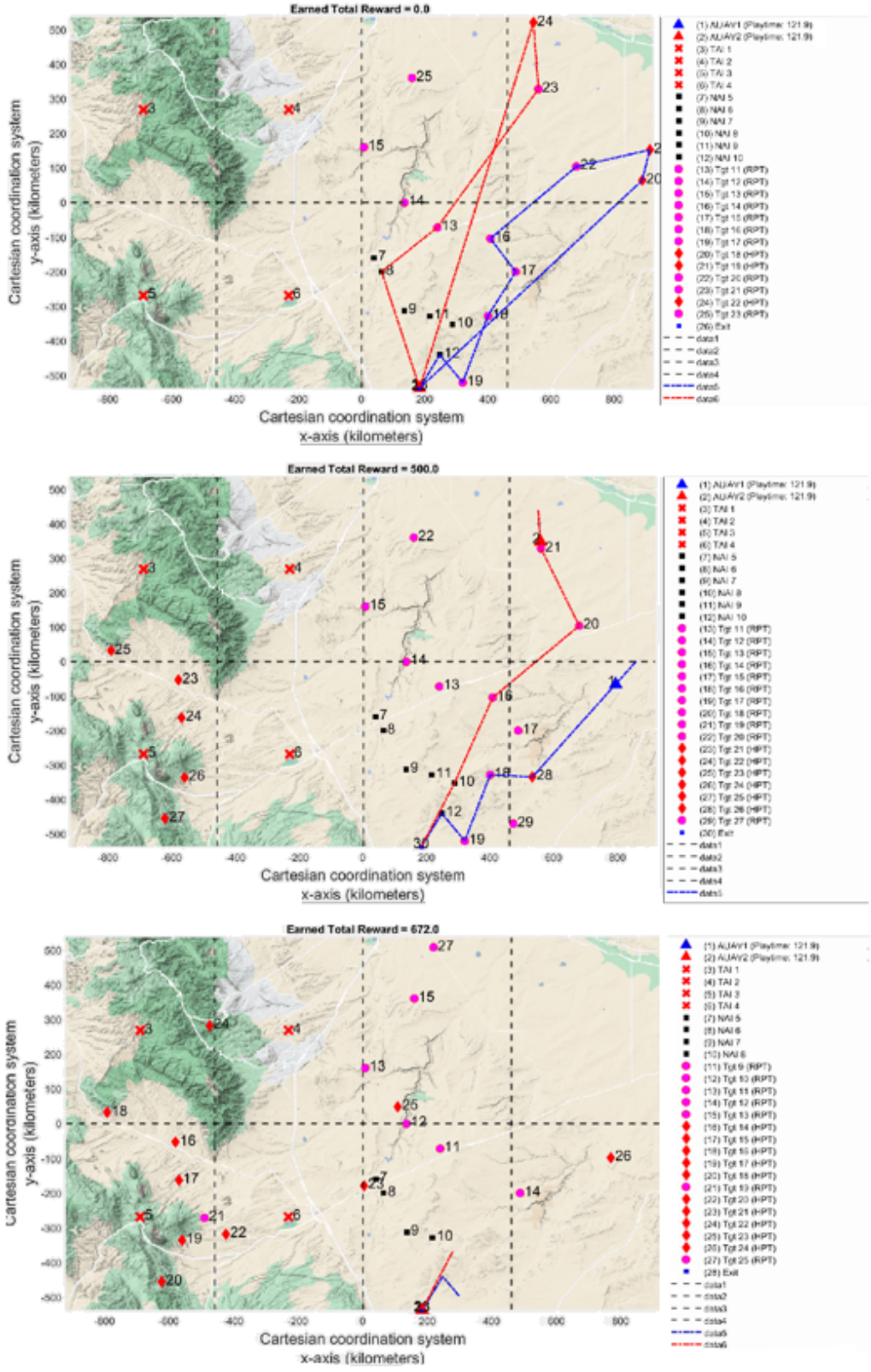

Figure 17. Sample Trajectory 698: π^{RGMH} (Benchmark) Policy

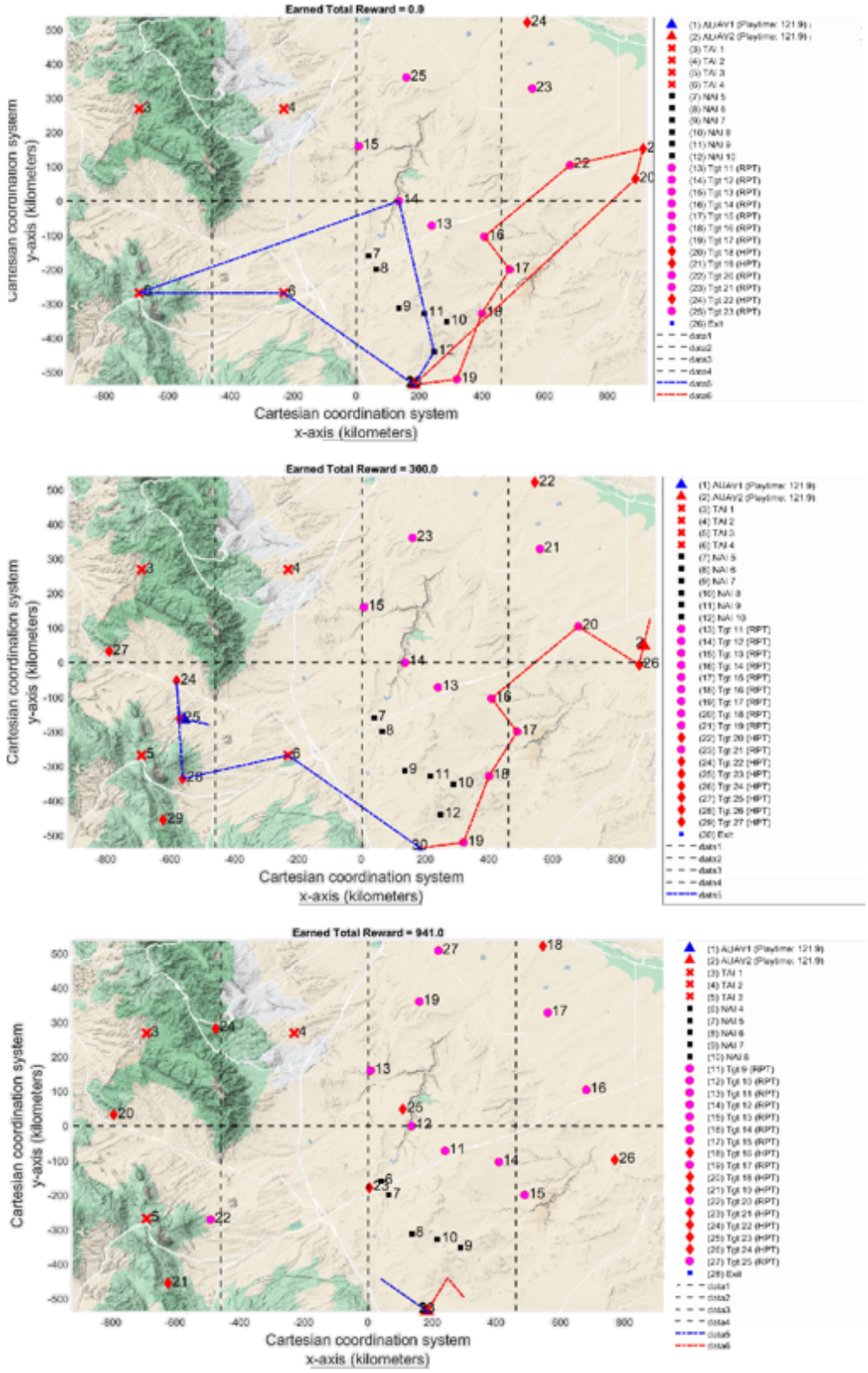

Figure 18. Sample Trajectory 698: $\pi^{CFA-RGMH}$ (ADP) Policy

The middle frame of Figure 18 shows that the AUAV makes an inefficient move to Node 25 and then proceeds to Node 24. Perhaps, had the AUAV moved to Node 24 before Node 25, there would have been sufficient time to add Node 29 to the JIPTL before fuel resources were expended. We note that this type of inefficient behavior may be resolved by an additional basis function that discourages it. Regardless, the ADP policy is able to achieve a total improvement of 269 points over the benchmark policy in this sample trajectory of the MRP-DTA.

This sample trajectory further supports the advantage of our ADP policy over the benchmark policy. The ADP policy learns a high-quality θ-value parameterization to apply to the base optimization model via the MADS algorithm. We display the θ-values in Figure 19 to further analyze the parameterization that is being applied to the base optimization model. We observe a θ_1-value of 80 units of reward applied to any target that arrives in between the playtime bounds of 121.46 minutes and 145.75 minutes. This augmentation applies to deliberate targets because they are assumed to have arrived with 145.75 minutes of playtime remaining. The soft bonus applies to both Node 5 and Node 6, which causes the blue AUAV to visit these TAIs early in its route. We distinguish that the parameterization no longer encourages each AUAV to move towards the top portions of the attack domain as was seen in Problem Instance 16. As playtime decreases, the parameterization continues to slightly alter the team behavior. Our CFA-DLA framework is able to learn high-quality behavior exhibited in the above parameterization given the problem features applied to Problem Instance 9.

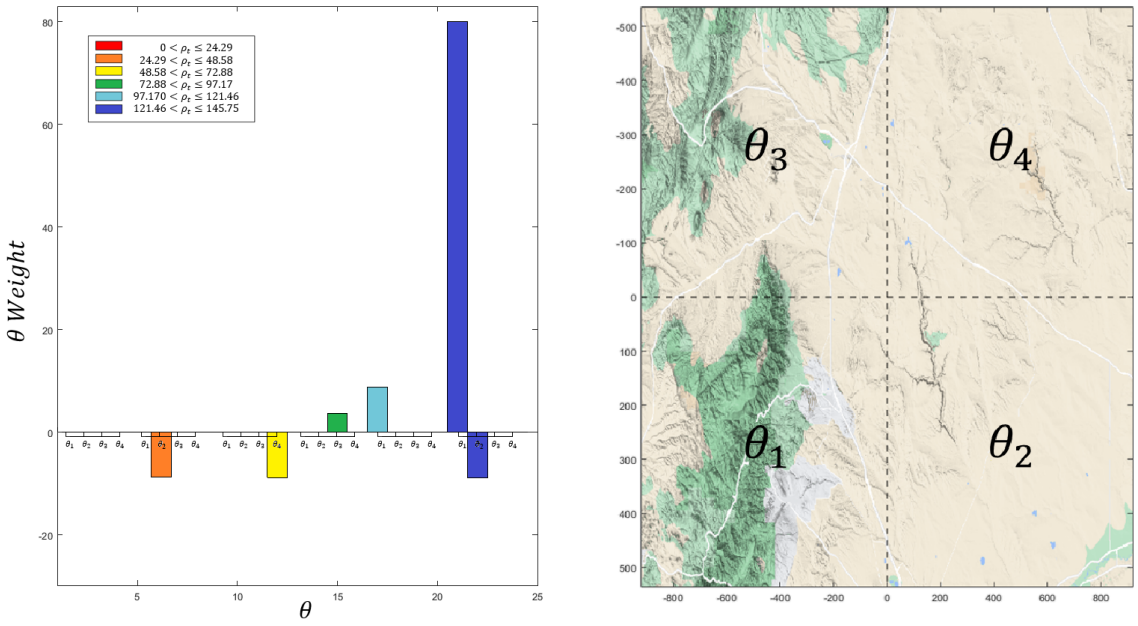

Figure 19. θ-values for Parameterization of ADP Policy for Problem Instance 9

4.5 RTOP Policy Excursion Analysis

The RTOP policy is the most computationally expensive benchmark policy under evaluation. We have found that the standard MRP-DTA baseline scenario is computationally intractable when using the RTOP policy solution procedure; therefore, we present a scalable excursion of the MRP-DTA where we remove targets from the scenario to reduce computational complexity and solve the resulting instance using the RTOP policy. We compare the performance of the RTOP benchmark policy against both the RSOP and RGMH policy as well as an ADP algorithm utilizing the RTOP policy, RSOP policy, and RGMH policy as the base optimization model. This scaled problem instance allows us to discuss the trade off between computational effort and quality of the base optimization model used within our CFA-DLA ADP algorithmic framework. We specifically wish to discuss and explore a value trade off that compares the solution quality of each policy versus the computational effort required to attach the policy.

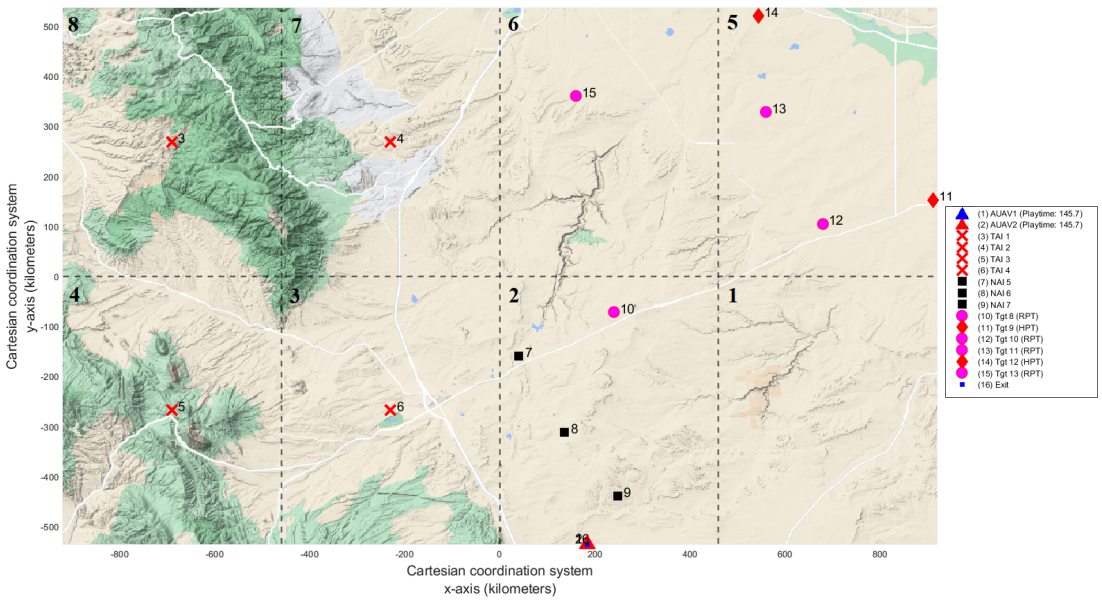

Figure 20. MRP-DTA Initialized Attack Domain in Matlab - Excursion Instance

The initial layout of targets in the attack domain can be seen in Figure 20. The targeting process has produced an ATO for 9 approved deliberate targets in the attack domain. Figure 11 fully depicts the starting scenario for all excursion scenarios. We choose to test only one problem feature setting of the MRP-DTA to allow for sufficient computational effort in tuning each ADP algorithm and discussion of the results. We tune the CFA algorithm for each base optimization model and compare against all benchmark policies. Table 17 displays all experimental factor levels, all fixed problem feature levels, and all fixed algorithm parameter levels.

The choice to focus experimental resources on the two parameters in Table 17 results from extensive preliminary testing on various algorithm parameters. These algorithm parameters directly impact the computational effort, which affected by the granularity of the tiling scheme, and raise compelling discussion regarding the trade off between added model granularity and computational effort. We evaluate each combination of algorithm parameters utilizing each base optimization model (i.e., RTOP policy, RSOP policy, and RGMH policy) to derive the set of algorithm

Table 17. MRP-DTA Problem Features

Experimental Parameters	**Factor Levels**
Basis Function Selection	Tile Only, Tile-Cluster, Tile-Cluster-Center
Discretization of Tiling Scheme	$2 \times 2 \times 6$, $2 \times 2 \times 8$, $4 \times 4 \times 4$
Fixed Problem Features	**Fixed Feature Level**
Target Arrival Rate	$\lambda = 0.10$
Probability of HPT Arrival	80%
Distribution of Arrivals	Left
AUAV Playtime (ρ_0)	145.75 minutes
Size of Attack Domain	1,840 km by 1,075 km
AUAV Speed	$16.466\bar{6}$ km/min
Number of NAIs	3 NAIs
Initial Number of Targets	9 HPTs, RPTs, NAIs
Initial Target Location	See Figure 11
Target Reward Values	See Table 5
Start/Exit Location	See Figure 11
Fixed Algorithm Parameters	**Fixed Algorithm Parameter Level**
Total Runtime	18,000 secs
Mesh Size Tolerance	1e-20
Initial Mesh Size	80 units
Inner Loop Size (M)	100

parameters that produces the highest solution quality for each ADP policy. Recall Table 10 for information on the experimental design.

Results show a statistically significant improvement in TR over the respective benchmark policy for each policy tested. The superlative policy found over all policies is found by implementing our CFA-DLA algorithm utilizing the RGMH policy as the base optimization model. It yields an improvement of 8.93% over the benchmark policy (π^{RGMH}). All results are shown in Table 18, which displays a 95% confidence interval over 1,000 simulated sample trajectories of the MRP-DTA for the problem instance of study. The results show that the computational efficiency of the base optimization model affects the ability to locate a high-quality policy shown by the superlative results found by $\pi^{CFA-RGMH}$. We further analyze Figure 21 to compare solution quality of all policies.

The RTOP benchmark policy is the most competitive benchmark policy among all benchmark policies with a TR of 493.72 points over 1,000 sample trajectories. Although this benchmark policy performs exceptionally well, $\pi^{CFA-RTOP}$ yields an inferior policy when compared to $\pi^{CFA-RGMH}$. This outcome is due to the computational complexity of implementing the RTOP policy to solve the MRP-DTA as we have recognized that the RTOP policy is the most computationally expensive policy under study. This means that our policy evaluation loop requires a much higher amount of computation time, significantly reducing the number of candidate policies that are tested.

Table 19 shows a comparison of 95% confidence intervals on computation times per for each sample trajectory of the MRP-DTA over all 1,000 sample trajectories. We report the 95% confidence intervals in terms of seconds. The results show that the RGMH policy requires significantly less computational time to calculate a sample trajectory of the applied MRP-DTA problem instance. This advantage allows for a

Table 18. Excursion Experimental Results

	RTOP Policy			
Design Point	**Mean**	**Halfwidth**	$\pi^{CFA-RTOP} - \pi^{RTOP}$	**% Improvement**
Benchmark	493.72	9.74	–	–
1 (226-T)	493.57	9.75	-0.15±0.65	-0.03 %
2 (226-TC)	497.50	9.77	3.79±3.49	0.77 %
3 (226-TCC)	493.30	9.80	-0.42±3.38	-0.09 %
4 (228-T)	495.96	9.72	2.25±3.48	0.46 %
5 (228-TC)	**498.74**	**9.68**	**5.02±3.66**	**1.02** %
6 (228-TCC)	494.84	9.66	1.13±4.47	0.23 %
7 (444-T)	495.85	9.74	2.14±3.04	0.43 %
8 (444-TC)	494.06	9.58	0.35±4.90	0.07 %
9 (444-TCC)	492.34	9.90	-1.38±5.00	-0.28 %
	RSOP Policy			
Design Point	**Mean**	**Halfwidth**	$\pi^{CFA-RSOP} - \pi^{RSOP}$	**% Improvement**
Benchmark	453.07	9.36	–	–
1 (226-T)	462.58	9.49	9.51±6.26	2.10 %
2 (226-TC)	470.84	9.34	17.77±6.46	3.92 %
3 (226-TCC)	467.75	9.65	14.68±6.64	3.24 %
4 (226-T)	**498.53**	**9.58**	**45.46±7.08**	**10.03** %
5 (226-TC)	462.68	9.28	9.61±6.55	2.12 %
6 (226-TCC)	488.78	9.70	35.71±7.12	7.88 %
7 (226-T)	483.51	9.42	30.44±6.97	6.72 %
8 (226-TC)	496.03	9.46	42.96±7.21	9.48 %
9 (226-TCC)	464.97	10.07	11.90±7.91	2.63 %
	RGMH Policy			
Design Point	**Mean**	**Halfwidth**	$\pi^{CFA-RGMH} - \pi^{RGMH}$	**% Improvement**
Benchmark	468.19	9.44	–	–
1 (226-T)	505.03	9.63	36.84±7.06	7.87 %
2 (226-T)	465.01	9.49	-3.19±7.25	-0.68 %
3 (226-T)	501.62	9.76	33.43±7.12	7.14 %
4 (226-T)	**510.01**	**9.58**	**41.82±6.86**	**8.93** %
5 (226-T)	501.38	9.35	33.19±6.81	7.09 %
6 (226-T)	498.59	9.76	30.40±7.01	6.49 %
7 (226-T)	482.76	9.80	14.56±6.75	3.11 %
8 (226-T)	488.75	9.72	20.56±7.15	4.39 %
9 (226-T)	485.48	9.92	17.29±7.71	3.69 %

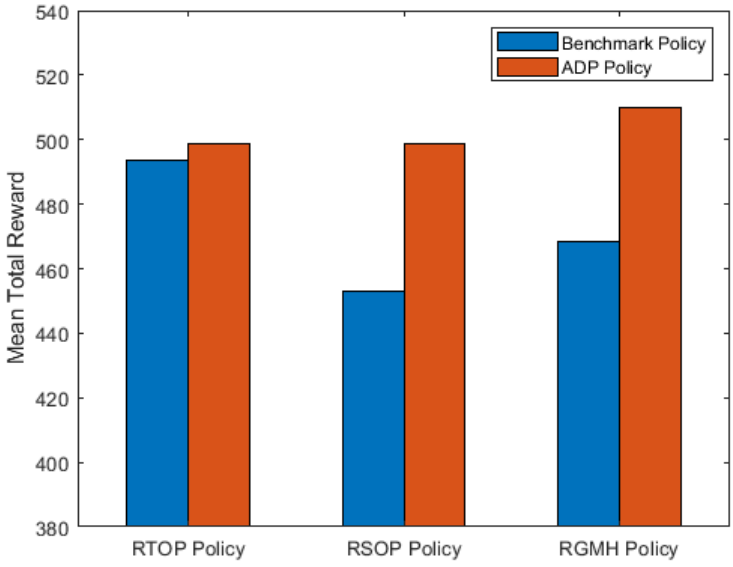

Figure 21. Solution Quality of All Policies over 1,000 Sample Trajectories

higher number of trial points (i.e., potential policies) to be tested. Although Figure 21 shows the RGMH benchmark policy is inferior to the RTOP benchmark policy in solution quality, the computational superiority of the RGMH benchmark policy produces the superior ADP policy when compared against all other policies. These results are critical in considering the tradeoff between solution quality and computational cost. We have shown that the highest quality base policy (i.e., the RTOP policy) does not provide the superlative CFA-DLA policy for this instance of the MRP-DTA due to its significant computational cost.

Table 19. 95% Confidence Intervals on Computation Times for all Base Policies (seconds)

Base Policy	Lower Level	Mean	Upper Level
RGMH Policy	0.0754	0.0768	0.0783
RSOP Policy	4.145	4.277	4.409
RTOP Policy	20.349	22.669	24.990

V. Conclusions and Future Recommendations

This research introduces a mutliagent routing problem that experiences dynamic target arrives throughout the problem execution. We formulate the sequential decision-making process utilizing a Markov decision process (MDP) modeling framework and further solve using a hybrid approximate dynamic programming (ADP) solution methodology. Our ADP framework utilizes a cost function approximation (CFA) to augment a direct lookahead (DLA) model with a parameterization that alters team behavior. We leverage a mesh adaptive direct search (MADS) algorithm to tune the parameterization of the base DLA optimization model and produce high-quality team attack policies.

We present a baseline problem instance of the multiagent routing problem with dynamic target arrivals (MRP-DTA) to test our CFA-DLA framework and further design an experiment to test 20 different problem feature settings. For each problem instance, we tune two separate algorithm parameters to determine the superlative ADP policy and report performance for comparison against our benchmark policies. We construct 95% confidence intervals on the difference in means to evaluate the margin of improvement achieved by the ADP policy over the benchmark policy and further discuss the factors that drive percent improvement over the benchmark policy. We perform two compelling case study evaluations to identify the benefits of utilizing our ADP framework. We then develop an excursion scenario that allows us to compare the performance of our most computationally expensive benchmark policy, the repeated team orienteering problem (RTOP) policy, with the repeated sequential orienteering problem (RSOP) policy and RGMH policy. Moreover, we compare these policies against a CFA-DLA policy obtained by using each policy as the base DLA optimization model. This excusrion analysis allows us to better understand the tradeoff between computational cost and solution quality of the base optimization

model and display the benefit of our solution approach.

5.1 Key Findings

The use of our CFA-DLA framework is successful in achieving anticipatory behavior and improving upon the benchmark policy used in the base optimization model. We have found that applying a parameterization to both the repeated greedy marginal heuristic (RGMH) and the RSOP policy base optimization models produces high-quality team attack policies for the MRP-DTA. We observe a statistically significant increase in performance for our CFA-DLA framework utilizing the RGMH policy base optimization model in 19 out of 20 problem instances with a 2.51% increase in performance for the baseline scenario of the MRP-DTA. When utilizing the RSOP policy as the base optimization model, the results show a statistically significant increase in performance for our CFA-DLA framework in 8 out of 10 problem instances with a 2.09% increase in performance for the baseline scenario of the MRP-DTA.

We discuss the robustness of our ADP policies and show that for 13 out of 20 problem instances the ADP policy outperforms the benchmark policy in total reward for the given number of sample trajectories utilizing the RGMH policy as the base optimization model. When utilizing the RSOP policy as the base optimization model, we show that for 10 out of 10 problem instances the ADP policy outperforms the benchmark policy in total reward for the given number of sample trajectories. These results help to inform stakeholders on the stability of the policy and its ability to return consistent, superior results over the benchmark.

5.2 Future Considerations

The premise of this research is to solve a stochastic air-to-ground USAF mission set utilizing a powerful ADP solution method by deriving high-quality attack policies that outperform comparable benchmark policies. We have shown that the magnitude of results are influenced by the problem features of the MRP-DTA and the algorithm parameters of our CFA-DLA solution approach. We contend further research on problem features and the applied solution procedure for the MRP-DTA would provide considerable insights to the USAF.

5.2.1 Problem Features

We have shown the effects of varying problem features when applying a CFA-DLA solution approach to the MRP-DTA. Specifically, we focus on testing variance in the stochastic elements of the system to inform decision-makers on the sensitivity of each stochastic feature and further distinguish instances where the USAF can significantly benefit from our CFA-DLA solution procedure. We do not heavily test other deterministic features of the MRP-DTA that may exhibit interesting results.

We recommend testing the initial layout of deliberate targets in the attack domain, to include the position of targets across the attack domain as well as the number and frequency of deliberate targets seen in the attack domain. This feature notably impacts the initial routing of AUAVs and potentially affects the marginal benefit of utilizing an ADP technique to induce anticipatory behavior. Furthermore, decision-makers may implement additional stochastic elements to the problem which would better mimic enemy disposition. A probability associated with a target's location or a probability associated with the number of co-located targets would instill relevant elements in the system and likely change and drive AUAV behavior.

We recommend implementing time-windows for strikes to more realistically rep-

resent the key aspects of the SCAR mission and better align with reality. This type of problem is known as the team orienteering problem with time windows (TOPTW) (Vansteenwegen *et al.*, 2009b). This imposes additional time constraints within which the team of AUAVs must schedule targets for destruction accordingly. Since AUAV behavior is heavily influenced by the characteristics of the problem, we believe this would affect the prioritization of targets in the attack domain and further reveal important problem features of the MRP-DTA that drive performance.

We critically assume that all targets are stationary during the mission to reduce the computational expense of tracking target motion models. Future research may consider the impact of tracking moving targets as it may impact the means by which AUAVs schedule targets for destruction. Furthermore, the addition of replenishment zones may provide intuition to stakeholders on valuable locations in which to establish forces for refueling and armament. By establishing replenishment zones within the attack domain, we would likely observe much different behavior wherein a single AUAV may localize to a region as opposed to traversing large areas of the attack domain to obtain large amounts of reward. Additionally, the USAF would benefit from the inclusion of armament on each AUAV as opposed to performing a strict sensory role for the AC-130U. The addition of an on-board AUAV resource for each AUAV adds an additional constraint to the TOP and increases the complexity of the problem while providing a more compelling role for the team of AUAVs.

5.2.2 Solution Procedures

We use a powerful CFA-DLA ADP technique to augment the base optimization model with a set of parameters that help to induce anticipatory behavior. We tune algorithm parameters and discuss the parameters that drive solution quality. We show via our excursion analysis the importance of reducing computation time given

the base optimization model whilst not sacrificing solution quality. Future research may investigate means to increase the solution quality of our proposed heuristic as the RGMH policy excels in computational cost but falls short with lower solution quality. Mechanisms may be implemented to increase the efficiency of the heuristic and produce a higher quality heuristic that eliminates suboptimal behavior. Vansteenwegen & Gunawan (2019) provide a general overview of various state of the art metaheuristics used to solve the team orienteering problem (TOP). Furthermore, other ADP techniques may produce high-quality solutions to the MRP-DTA. Although we have shown that the CFA-DLA approach is a valid and robust approach to solve for a high-quality ADP policy, future research may evaluate other successful techniques for their reduction in computational burden, improvement in the quality of behavior, or superior robustness.

Subsequent research may evaluate the effectiveness of a value function approximation (VFA) ADP approach. Past research has studied comparable problems to the MRP-DTA and implemented a VFA approach known as approximate policy iteration, specifically using a least-squares temporal difference (API-LSTD) update to the value function. Although the solution approach did not show definitive improvement over the benchmark policy for the problem of study, we believe that VFA is a powerful algorithmic framework that can be tuned to suit the MRP-DTA. The tuning of the algorithm becomes a time consuming, yet crucial element of the process that inform the convergence to a high-quality solution. If the tuning is performed correctly, the resulting policy has the potential to handle specific events in the MRP-DTA that are handled inefficiently when utilizing a CFA-DLA approach. We recommend utilizing neural network regression, which is a powerful and robust framework for updating the value function approximation and potentially provides improved results at a notable computational cost. Furthermore, one might investigate a hybrid DLA-VFA ADP ap-

proach similar to the solution method used in Ulmer *et al.* (2019). An online rollout algorithm helps to induce anticipatory behavior in a problem setting that experiences dynamic target arrivals by rolling out the horizon of the problem and estimating the impact of decisions. The remaining VFA portion of the algorithm incorporates estimates of the state-action pair to help inform decisions moving forward. This result provides a potential solution method to the MRP-DTA, utilizing a powerful solution method that is less visible in the ADP literature.

Future research considerations potentially yield significant results, which aid in the development of autonomous systems. The utility of AUAVs as FOs in the SCAR mission provide the USAF with great potential to bolster air superiority. Holistically, we intend for the results and analysis of the MRP-DTA and the accompanying CFA-DLA ADP solution method to inform the development of autonomous systems in future USAF combat roles.

Bibliography

Air Force Technology. 2003. *X-45 J-UCAV (Joint Unmanned Combat Air System)*. `https://www.airforce-technology.com/projects/x-45-ucav/`.

Audet, Charles, & Dennis Jr, John E. 2006. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, **17**(1), 188–217.

Bain, Lee J, & Engelhardt, Max. 1992. *Introduction to probability and mathematical statistics*. Vol. 4. Duxbury Press, Belmont, CA.

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

Bayliss, Christopher, Juan, Angel A, Currie, Christine SM, & Panadero, Javier. 2020. A learnheuristic approach for the team orienteering problem with aerial drone motion constraints. *Applied Soft Computing*, **92**, 106280.

Brown Jr, Gen Charles. 2020. *Accelerate Change or Lose*. United States Air Force, Washington D.C., United States.

Brunson, Maj B. Trent. 2007. *Air-Ground Integration*. Vol. 2. Air Land Sea Application Center.

Cahoon, Lt Col Troy. 2021. *Technology in Peacekeeping: How Technology Can Make Peacekeeping Better*. Air University.

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

Department of Defense. 2016. *Joint Publication 3-03: Joint Interdiction*. Washington, D.C.

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

Department of Defense. 2017b. *Joint Publication 3-01: Countering Air and Missile Threats*. Washington, D.C.

Department of Defense. 2017c. *Joint Targeting School Student Guide*. Washington, D.C.

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

Department of Defense. 2019b. *Joint Publication 3-31: Joint Land Operations*. Washington, D.C.

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

Department of Defense. 2021. *Department of Defense Dictionary of Military and Associated Terms*. Washington, D.C.

Department of the Air Force. 2019a. *Air Force Doctrine Publication 3-01: Counterair Operations*. Washington, D.C.

Department of the Air Force. 2019b. *Air Force Doctrine Publication 3-52: Airspace Control*. Washington, D.C.

Department of the Air Force. 2019c. *Air Force Doctrine Publication 3-60: Targeting*. Washington, D.C.

Department of the Air Force. 2019d. *U.S. Air Force 2030 Science and Technology Strategy*. Washington, D.C.

Department of the Air Force. 2020. *Air Force Doctrine Publication 3-03: Counterland Operations*. Washington, D.C.

Department of the Air Force. 2021. *Air Force Doctrine Publication 1: The Air Force*. Washington, D.C.

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

Ghadimi, Saeed, Perkins, Raymond T, & Powell, Warren B. 2020. Reinforcement Learning via Parametric Cost Function Approximation for Multistage Stochastic Programming. *arXiv preprint arXiv:2001.00831*.

Golden, Bruce L, Levy, Larry, & Vohra, Rakesh. 1987. The orienteering problem. *Naval Research Logistics (NRL)*, **34**(3), 307–318.

Grindle, Charles, Lewis, Michael, Glinton, Robin, Giampapa, Joseph, Owens, Sean, & Sycara, Katia. 2004. Automating Terrain Analysis: Algorithms for Intelligence Preparation of the Battlefield. *Proceedings of the Human Factors and Ergonomics Society*.

Haider, Lieutenant Colonel André. 2019. A Comprehensive Approach to Countering Unmanned Aircraft Systems. *Joint Air Power Competence Centre (JAPCC)*.

Hicks, Maj Gen (S) J. Marcus. 2014. Spirit 03 and the Golden Age of the AC-130 Gunship. *Air Commando Journal*, **3**, 30–34.

Hosseini, Seyyed Soheil Sadat, Jafarnejad, Ali, Behrooz, Amir Hossein, & Gandomi, Amir Hossein. 2011. Combined heat and power economic dispatch by mesh adaptive direct search algorithm. *Expert Systems with Applications*, **38**(6), 6556–6564.

Hubmann, Constantin, Becker, Marvin, Althoff, Daniel, Lenz, David, & Stiller, Christoph. 2017. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. *Pages 1671–1678 of: 2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE.

Ismail, Adiel, Tuyishimire, Emmanuel, & Bagula, Antoine. 2018. Generating Dubins path for fixed wing UAVs in search missions. *Pages 347–358 of: International Symposium on Ubiquitous Networking*. Springer.

Javed, M Yaqoob, Murtaza, Ali Faisal, Ling, Qiang, Qamar, Shahid, & Gulzar, M Majid. 2016. A novel MPPT design using generalized pattern search for partial shading. *Energy and Buildings*, **133**, 59–69.

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

Jenkins, Phillip R, Robbins, Matthew J, & Lunday, Brian J. 2021b. Approximate dynamic programming for the military aeromedical evacuation dispatching, preemption-rerouting, and redeployment problem. *European Journal of Operational Research*, **290**(1), 132–143.

Jeong, Byeong-Min, Ha, Jung-Su, & Choi, Han-Lim. 2014. MDP-based mission planning for multi-UAV persistent surveillance. *Pages 831–834 of: 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*. IEEE.

Li, Ming, Bai, He, & Krishnamurthi, Niyant. 2019. A Markov Decision Process for the Interaction between Autonomous Collision Avoidance and Delayed Pilot Commands. *IFAC-PapersOnLine*, **51**(34), 378–383.

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

Meilinger, Phillip S. 2014. *Air Interdiction*. Air Force Association.

Panadero, Javier, de Armas, Jesica, Currie, Christine SM, & Juan, Angel A. 2017. A simheuristic approach for the stochastic team orienteering problem. *Pages 3208–3217 of: 2017 Winter Simulation Conference (WSC)*. IEEE.

Papapanagiotou, Vassilis, Montemanni, Roberto, & Gambardella, Luca Maria. 2015. The Orienteering Problem with Stochastic Travel and Service: Times New approaches to sampling-based objective function evaluation. *Page 75 of: International Conference on Computational Mathematics, Computational Geometry & Statistics (CMCGS). Proceedings*. Global Science and Technology Forum.

Pěnička, Robert, Faigl, Jan, & Saska, Martin. 2019. Physical orienteering problem for unmanned aerial vehicle data collection planning in environments with obstacles. *IEEE Robotics and Automation Letters*, **4**(3), 3005–3012.

Perkins, Raymond T, & Powell, Warren B. 2017. Stochastic optimization with parametric cost function approximations. *arXiv preprint arXiv:1703.04644*.

Pichpibul, Tantikorn, & Kawtummachai, Ruengsak. 2013. A heuristic approach based on clarke-wright algorithm for open vehicle routing problem. *The Scientific World Journal*, **2013**.

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

Powell, W. 2022. *Reinforcement Learning and Stochastic Optimization*. John Wiley & Sons, Hoboken, NJ. Draft.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2nd edition, Hoboken, NJ.

Puterman, Martin L. 1994. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Hoboken, NJ.

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

Robbins, Matthew J, Jenkins, Phillip R, Bastian, Nathaniel D, & Lunday, Brian J. 2020. Approximate dynamic programming for the aeromedical evacuation dispatching problem: Value function approximation utilizing multiple level aggregation. *Omega*, **91**, 102020.

Sayler, Kelley M. 2020. *Defense Primer: U.S. Policy on Lethal Autonomous Weapon Systems*. Congressional Research Service. `https://crsreports.congress.gov/product/pdf/IF/IF11150`.

Secomandi, Nicola. 2001. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, **49**(5), 796–802.

Shuai, Hang, Fang, Jiakun, Ai, Xiaomeng, Yao, Wei, Wen, Jinyu, & He, Haibo. 2019. On-line energy management of microgrid via parametric cost function approximation. *IEEE Transactions on Power Systems*, **34**(4), 3300–3302.

Sundar, Kaarthik, Venkatachalam, Saravanan, & Rathinam, Sivakumar. 2016. Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem. *Pages 6489–6494 of: 2016 American Control Conference (ACC)*. IEEE.

Thakoor, Omkar, Garg, Jugal, & Nagi, Rakesh. 2019. Multiagent UAV routing: A game theory analysis with tight price of anarchy bounds. *IEEE Transactions on Automation Science and Engineering*, **17**(1), 100–116.

Thayer, Thomas C, & Carpin, Stefano. 2020. Solving large-scale stochastic orienteering problems with aggregation. *Pages 2452–2458 of: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Thayer, Thomas C, & Carpin, Stefano. 2021. An Adaptive Method for the Stochastic Orienteering Problem. *IEEE Robotics and Automation Letters*.

Ulmer, Marlin W, Goodson, Justin C, Mattfeld, Dirk C, & Hennig, Marco. 2019. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, **53**(1), 185–202.

Ulmer, Marlin W, Goodson, Justin C, Mattfeld, Dirk C, & Thomas, Barrett W. 2020. On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics*, **9**(2), 100008.

USAF. 2021. *USAF Fact Sheet: AC-130U*. `https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104486/ac-130u/`.

Vansteenwegen, Pieter, & Gunawan, Aldy. 2019. *Orienteering Problems*. Springer, Cham, Switzerland.

Vansteenwegen, Pieter, Souffriau, Wouter, Berghe, Greet Vanden, & Van Oudheusden, Dirk. 2009a. A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research*, **196**(1), 118–127.

Vansteenwegen, Pieter, Souffriau, Wouter, Berghe, Greet Vanden, & Van Oudheusden, Dirk. 2009b. Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, **36**(12), 3281–3290.

Wallace, Nathan D, Kong, He, Hill, Andrew J, & Sukkarieh, Salah. 2020. The orienteering problem with replenishment. *Pages 973–978 of: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE.

Wilson, Heather. 2019. *US Air Force Science and Technology Strategy 2030 and Beyond*. `https://www.af.mil/Portals/1/documents/2019%20SAF%20story%20attachments/Air%20Force%20Science%20and%20Technology%20Strategy.pdf`.

Zhang, Shu, Ohlmann, Jeffrey W, & Thomas, Barrett W. 2018. Dynamic orienteering on a network of queues. *Transportation Science*, **52**(3), 691–706.

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	September 2020 — March 2022

4. TITLE AND SUBTITLE

Multiagent Routing Problem with Dynamic Target Arrivals Solved via Approximate Dynamic Programming

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Mogan, Andrew E, 2nd Lt, 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-156

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Strategic Development Planning & Experimentation (SDPE) Office
Mr. David M. Panson
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 This research formulates and solves the multiagent routing problem with dynamic target arrivals (MRP-DTA), a stochastic system wherein a team of autonomous unmanned aerial vehicles (AUAVs) executes a strike coordination and reconnaissance (SCAR) mission against a notional adversary. Dynamic target arrivals that occur during the mission present the team of AUAVs with a sequential decision-making process which we model via a Markov Decision Process (MDP). To combat the curse of dimensionality, we construct and implement a hybrid approximate dynamic programming (ADP) algorithmic framework that employs a parametric cost function approximation (CFA) which augments a direct lookahead (DLA) model via a parameterization to the objective function. We show a statistically significant improvement over the repeated greedy marginal heuristic benchmark policy for 19 out of 20 problem instances and a statistically significant improvement over the repeated sequential orienteering problem benchmark policy for 8 out of 10 problem instances of the MRP-DTA. Results of excursion analysis show the value tradeoff of balancing solution quality and computational effort when selecting the base optimization model for our CFA-DLA algorithm.

15. SUBJECT TERMS

approximate dynamic programming, reinforcement learning, artificial intelligence, autonomous attack aviation, route planning, targeting, team orienteering problem, strike coordination and reconnaissance mission

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 J. Robbins, AFIT/ENS
U	U	U	UU	130	19b. TELEPHONE NUMBER *(include area code)* (937)255-3636,x4606; matthew.robbins@afit.edu