ICAS 2020 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

QUANTIFICATION OF OPERATIONAL EFFECTIVENESS FOR TECHNOLOGY EVALUATION

Andrew Hull[1], Alicia Sudol[1] & Dimitri Mavris[1]

[1]Georgia Institute of Technology

Abstract

Modern methods for technology evaluation lack proper quantification of operational effectiveness. Mission simulations must move away from a fixed or constrained mission model to one that is open and capable of exploring tactics for a new technology. The framework presented herein proposes an agent-based decision behavior approach for such tactics exploration using deep reinforcement learning. The novel classification of the agent's decision behavior algorithm paired with multiple state behavior algorithms enables a broader exploration of the mission action design space. An unmanned aerial vehicle conducting wildfire surveillance and suppression is chosen as a proof-of-concept mission. This approach demonsrates the quantification of operational effectiveness within a minimally defined mission scenario.

Keywords: Reinforcement Learning, Operations Analysis, Technology Evaluation, Agent-Based Modeling and Simulation

1. Introduction

Wildfire surveillance and suppression can be a dangerous and expensive task for manned aircraft. The U.S. Forest Service and Department of Interior spent on average over 2.3 billion dollars on wildfire suppression per year from 2015-2020 [1]. Over a quarter of firefighter fatalities are aviation related due to the high-risk, low-altitude environment [2]. Unmanned Aerial Vehicles (UAVs) can provide a safe and cost effective platform to aid firefighters in wildfire suppression. The UAV platform is of particular interest due to its real-time high-resolution imaging that can inform firefighters with critical information to help contain the fire. Their capabilities have been demonstrated through the NASA Ikhana unmanned aircraft flight tests [3].

As the surveillance capabilities of UAVs increase with new technologies, their impact on wildfire suppression should also be studied. Technology evaluation methods could be utilized to inform technology investment decisions that result in safer and more effective aerial firefighting. Emerging technology evaluation methodologies such as Technology Identification, Evaluation, and Selection for System of Systems [4] or Effectiveness-Based Design [5] seek to improve the quantification of mission effectiveness during technology evaluation. These methodologies seek to quantify the *means* – the technologies used to perform a mission – and the *ways* – the tactics used to complete a mission – of the technology design space. The means for a firefighting UAV include things such as the aircraft's optics, payload capacity, or payload refilling method. The impact of alternative means on mission performance can be quantified using existing methods. The ways, which can be more difficult to quantify, refers to how an aircraft conducts its mission, such as flight path or order of actions.

Quantifying the ways is traditionally conducted late in the design process with a fixed vehicle; however, if performance requirements are not satisfied at this point, there is little to no room left to iterate on the design [6]. A new approach that can quantify a technology's mission effectiveness with no preset mission path, platforms, or envelope early in the design process is desired [5]. Proper evaluation of the ways would provide critical information to the decision-maker during technology selection.

The mission framework presented herein enables the exploration of the ways a technology could be used to augment the quantification of mission effectiveness during technology evaluation. This agent-based decision framework is implemented for an autonomous aircraft conducting wildfire surveillance and suppression to aid firefighters in wildfire containment. Such a mission will provide a proof-of-concept for the proposed agent-based decision framework.
The first section outlines the selection of agent -based modeling and the appropriate agent framework. Section 3identifies a decision making algorithm to integrate with a decision and state behavior approach. The decision behavior algorithm is implemented for a wildfire surveillance and suppression proof-of-concept mission in Section 4.

2. Framework Definition

Current methods to explore the optimal tactical decisions made during a mission scenario for a given technology are inadequate due to the predetermined nature of mission scenarios. The modeling framework desired to explore optimal decisions – also known as the mission action design space – is one that defines no preset mission path, platforms, or envelope [5]. The difficulty in achieving this mission simulation environment presents itself through traceability and dimensionality. The optimal mission actions for a given technology must be traceable to ensure that the quantified operational effectiveness is believable to the decision-maker. The dimensionality must also be addressed to ensure that the proposed framework can explore many tactical combinations for each unique technology under investigation. Defining a model that can explore the mission action design space requires a more targeted framework to enable the formulation and exploration of tactics.
The first step in framework definition is identifying the appropriate modeling method. The modeling and simulation (M&S) environment traditionally used to simulate mission outcomes can range from system dynamics, discrete event simulation (DES), or agent-based modeling (ABM) [4]. ABM can often be defined using three main elements: the active entities or agents, the relationships between agents, and the agent's interaction with their environment [9]. The definition of an agent's behavior with its environment requires no knowledge of the overall behavior the system may produce. Such a model formulation would enable mission simulations without defining a mission path or envelope a priori. The decentralized control found in ABM aligns more closely with the exploration of the mission action design space than centralized control found in DES. The agent's ability to explore its environment and make its own decisions without a central control is critical in enabling the discovery of effective mission actions [8]. For these reasons, ABM is selected as the appropriate modeling framework.

2.1 Agent-Based Framework

There are many agent framework formulation methods available within ABM. These methods provide a range of definitions of behavioral rules for the agent to follow. State machines are one way to model agents in an ABM. This approach focuses on the possible states an agent can be in rather than traversing a set of discrete decision branches in a decision tree. The decision-making aspect of state machines comes into play when formulating the criteria required to transition states. State transitions can occur given a user defined decision or a reaction to stimuli [7]. The ability to predefine each state with a template of predetermined state transitions significantly reduces the complexity of each behavior by isolating the transitions in and out of a state [10]. These characteristics of state machines can also be found in a sub-category of state machines known as Markov Decision Processes (MDPs). MDPs explicitly define the states of the system and aim to better define the relationship between microscopic and macroscopic properties [11, 12]. This mapping of microscopic and macroscopic properties would provide traceability of decisions made throughout a mission. The MDP framework, illustrated in Figure 1, consists of a set of finite states, S, which define the system or environment, a finite set of possible actions, A, to choose from, a state-transition function, T, and a reward function, R. The agent uses the state-transition and reward functions to determine the policy, $\pi(S_t)$. The agent's policy chooses an action, A_t, given a state, S_t, that provides the highest expected reward R_t. Once an action is selected, the agent performs the action, receives a reward R_{t+1}, for that action, and transitions to a new state, S_{t+1}. The optimal set of decisions is one that maximizes the rewards. This type of framework is similar to those used in traditional Artificial Intelligence (AI) systems which plan

a set of sequential actions. However, a plan may rarely execute as expected in an uncertain system or environment. In order to create a more open mission action design space, an AI planning model that can operate in a stochastic environment is desired.

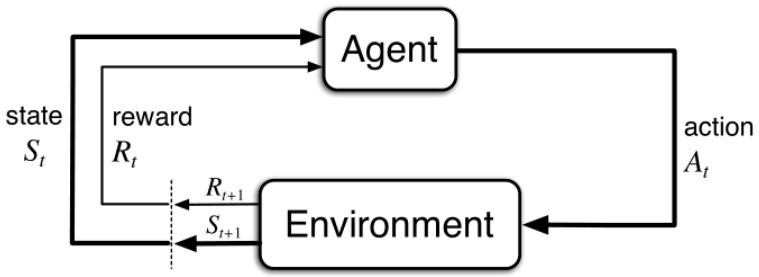

Figure 1 – Markov Decision Process of an agent interacting with its environment [13].

MDPs with imperfect information differ from traditional MDPs and state machines due to the introduction of stochasticity into the transition criteria between each state [12, 14]. This allows the agent to react to an uncertain state. MDPs with imperfect information could be useful in defining a system or ABM where not all state variables are known [15]. This class of MDP is formally known as a Partially Observable MDP (POMDP) [16]. POMDPs are defined similarly to MDPs except they include an additional observation function. The observation function defines the probability of an observation from a set of finite observations given action, a, and chosen state, s'. The ability of an agent to use observations to estimate its state would enable further exploration of the mission action design space through the ability to reevaluate its actions based on changing observations of the environment.

2.2 Decision Behavior Framework

The next step in formulating a state-based agent framework is to identify a method that can allow the agent to alter its decision preferences to change in observations of the environment. In order to balance the traceability of the framework with minimal definition of the mission structure, the mission action design space can be discretized into decisions and resulting states. The decisions made throughout the mission can be defined as the decision behavior of the agent while the actions taken between decisions can be defined as the state behaviors. The individual states an agent can choose from are each unique behaviors with different actions and goals. This formulation of decision and state behaviors favors a minimal discretization of the mission into a finite set of critical decision points.

Each agent behavior should be of a functional form to allow for observed stimuli to impact the agent's decisions. In order to determine the functional form of the decision and state behavior functions, an algorithm must be identified that can solve for the optimal behavior. From this perspective, a decision behavior algorithm can be defined as an agent's internal decision-making preferences that are a function of stimuli from its environment. The agent's state behavior algorithms should be defined similarly; however the internal decisions made within a state behavior algorithm will impact specific agent actions while the decisions made within a decision behavior algorithm impact the available actions to the agent.

For wildfire surveillance and suppression, state behaviors include the agent's ingress into the impacted region, wildfire surveillance, wildfire suppression, refilling the agent's payload from the nearest open water source and the egress path out of the impacted region. One critical decision point in this mission is the decision to continue to survey the fire front to gain more information on its movement or fight the fire using information gained from observations. The combination of decision and state behaviors over an entire mission will produce the collective mission behavior of a single agent in a more traceable way. These three classifications of agent behavior are illustrated in Figure 2.

The introduction of decision and state behavior algorithms provides a way to trace the behavior of an agent throughout a mission simulation. This discretization of the mission action design space also reduces the dimensionality of an agent by placing some limitations on the number of available states and actions for each state behavior. This decision and state behavior algorithms provides a

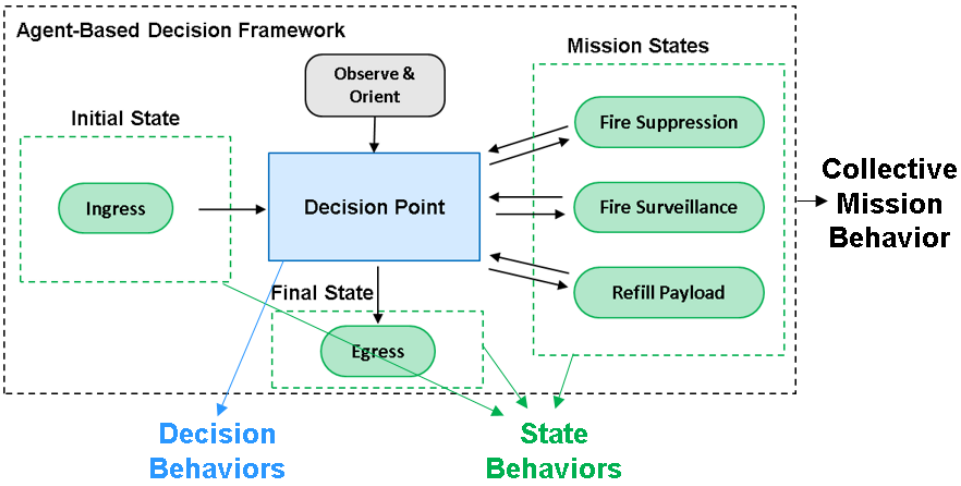

Figure 2 – The decision behavior, state behavior, and collective mission behavior of an agent-based decision framework.

traceable way to explore the mission action design space while providing a structure that addresses its dimensionality.

2.3 Algorithm Selection

The decision behavior algorithm defines the decision-state mapping, while the state behavior algorithms define each behavior's unique state-action mapping. This mapping requires an algorithm that can explore and learn from the mission action design space. Reinforcement Learning (RL) is one algorithm framework in particular that learns through interactions with its environment by trial and error [13]. This framework focuses on algorithms that map stimuli to actions. This type of learning algorithm is utilized when the agent must discover the optimal behavior through rewards in an environment where the rules are unknown. Such an algorithm is suitable for exploring the mission action design space due to its unknown shape and complexity. The standard formulation of an RL agent is illustrated in Figure 3. Similar to the MDP structure, an agent learns a behavior, B, by taking an action, A, enters a new state, S, and receives a reward, R. The input function, I, generalizes the framework for perfect and imperfect information formulations.

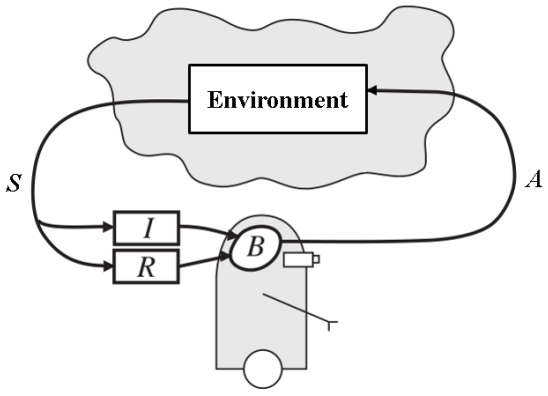

Figure 3 – General framework of an RL agent where A is the action chosen by the agent, S is the agent's state, R is the reward received, I is the input information, and B is the agent's behavior. Modified from [17].

The applicability of RL on problems with perfect and imperfect information of its environment enable its use across a variety of missions. For wildfire surveillance, the fire front is constantly moving which requires the continuous update of information to effectively contain it. This lack of global information requires the agent to make observations and corresponding decisions based on imperfect information

of the fire front. Both the decision and state behavior algorithms can utilize a POMDP framework with RL to enable broad applicability to many mission types.

2.4 Mapping Decisions, States, and Actions

The state definition of an agent can vary in complexity depending on the amount of data or information required to capture the decision-state or state-action mapping. Both mappings use a unique set of continuous state variables along with an observed image of its environment that define the agent's current information. These state variables are then mapped to a ranked set of actions the agent believes would provide the highest reward. For a decision-state mapping, the set of actions are a discrete set of available state behaviors at each decision point. The state-action mapping links the agent's state variables and observations to a discrete set of actions, such as flight control parameters, chosen for every time step. The dimensionality of both the decision-sate and state-action mapping for continuous imperfect information problems can quickly explode in size. This drives the need to use a function approximation to map the large and unknown space.

Artificial Neural Networks (ANNs) are function approximators that have been used extensively in ABM and RL [7, 18, 19]. The structure of the model is comprised of many artificial neurons that link inputs to outputs based on their connections within the hidden layers of the algorithm. Each ANN can have any number of hidden layers depending on the complexity of the model desired. The artificial neurons are simple processing units that are interconnected in an intentional way [19]. Each connection has an associated weight that determines its impact on its neighboring neurons. The weights of each neuron are adjusted during learning.

The ANN would replace the traditional discrete table mapping by providing a complex network mapping inputs to outputs. The advantage of using these models would be the ability to include continuous state variables and image observations for the agent's input information such as those implemented in [18]. The use of ANNs for both the decision-state and state-action mapping provides a method to implement the decision and state behavior algorithms.

3. Decision Behavior Approach

The decision and state behavior algorithms propose an approach to address the dimensionality of the mission action design space by defining the mission as a set of sequentially chosen state behaviors and addresses the desired traceability of such a framework through an agent's decision behavior. This decision behavior approach is defined by five steps, illustrated in Figure 4.

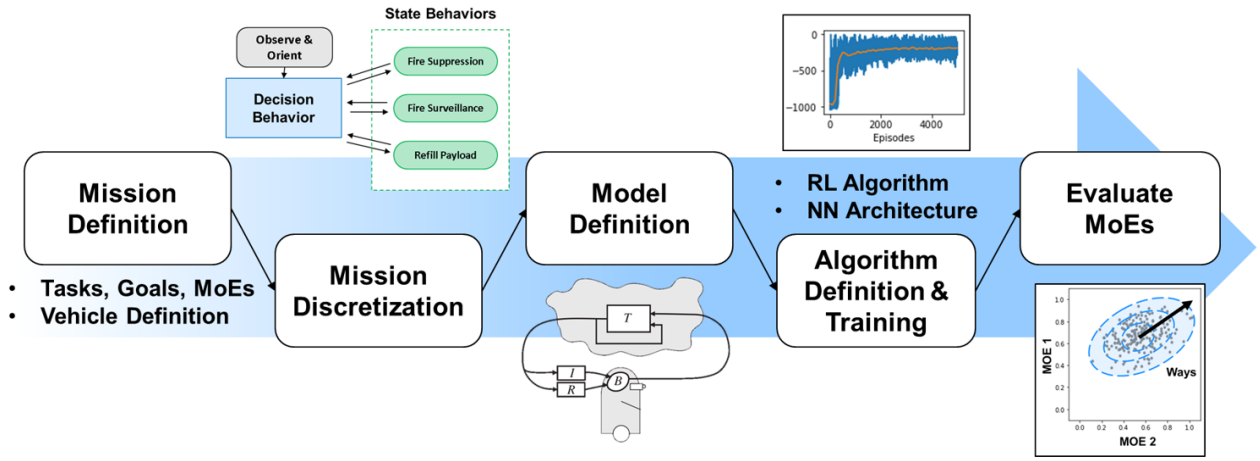

Figure 4 – The five step decision behavior approach.

The first step is to define the mission by identifying the tasks and goals of the mission. This step should outline the general mission structure without defining any mission paths or envelopes. The second step breaks the mission down into discrete decision points and state behaviors. Each state behavior could consist of a single task or grouping of tasks, depending on the task complexity. The third step develops the ABM using a POMDP framework that incorporates imperfect information such

as agent observations. This step also defines the ANN structure that will map the state variables and agent observations to the agent's available actions.
Step four defines the RL algorithm and trains the decision and state behaviors. This approach does not specify the type of RL algorithm required for either of the decision or state behaviors due to the ever evolving field of RL. The user of this approach should weigh their familiarity of an RL algorithm with the most recent advancements in literature.
Training of the RL algorithms will determine the state behavior's solution first using a representative mission segment. For wildfire surveillance, a mission segment could be one where a fire is randomly propagated for a finite amount of time. Once the fire has propagated, the agent will spawn randomly on the map, find the fire front and update its knowledge through observations for a given amount of time. The decision behavior algorithm can then be trained using the converged state behavior algorithms to traverse through the entire mission scenario. The number of decision points within a given mission may vary depending on the criteria that triggers the decision behavior algorithm or defines the end of the mission.
The final step in this decision behavior approach is the quantification of mission effectiveness. The trained decision algorithm should be simulated in a stochastic environment that allows for variability in the mission outcome. Measures of Effectiveness (MoEs) should then be quantified over many repititions to determine the mission effectiveness of the agent.
This mission modeling approach could be extended to multi-capability platforms as well. The capability of a technology could be quantified by measuring its mission effectiveness across multiple missions each with a unique decision behavior algorithm while reusing the same state behaviors. The extension of a multi-mission decision behavior approach is not addressed in this work.

4. Proof of Concept

The decision behavior approach is demonstrated for an autonomous aircraft conducting wildfire surveillance and management. The goal of this example mission is to highlight the ability of the decision behavior approach, given a minimally defined mission, to quantify an agent's mission effectiveness while providing a traceable way to investigate its collective behavior. The results of this approach will inform its applicability to future use with technology evaluation methodologies.

4.1 Mission Definition

The mission selected for this proof-of-concept will investigate the effectiveness of a single UAV that can conduct wildfire surveillance and suppression. The tasks for this mission include fire surveillance, fire suppression, and payload refill. Simplification of the problem is made by assuming the UAV maintains a constant altitude and speed throughout the mission.

4.2 Mission Discretization

The mission for this proof-of-concept will have three state behaviors which include wildfire surveillance, fire suppression, and payload refilling. Wildfire surveillance seeks to increase the agent's knowledge of the fire for a fixed amount of time. Fire suppression flies to an optimal location chosen by the agent and drops water on that portion of the fire. The last state behavior seeks to refill its payload by flying to the closest available water source. For this mission, the ingress and egress of the agent are not modeled and are assumed to be fixed.
The critical decision points for this mission will arise once a state behavior has met its completion criteria. For example, once the wildfire surveillance behavior observes the fire for 100 time steps, the decision behavior point will be triggered. This decision point will have to decide if it should continue surveillance of the fire, conduct fire suppression, or refill its payload with water. This mission logic is illustrated in Figure 5.

4.3 Model Definition

The ABM defined in this section investigates an aircraft flying at constant altitude with a speed of $20km/s$ in a $1km^2$ region. The aircraft dynamics and wildfire model used are defined in [18]; however the state, observations, actions, and rewards are defined differently for this study. The decision

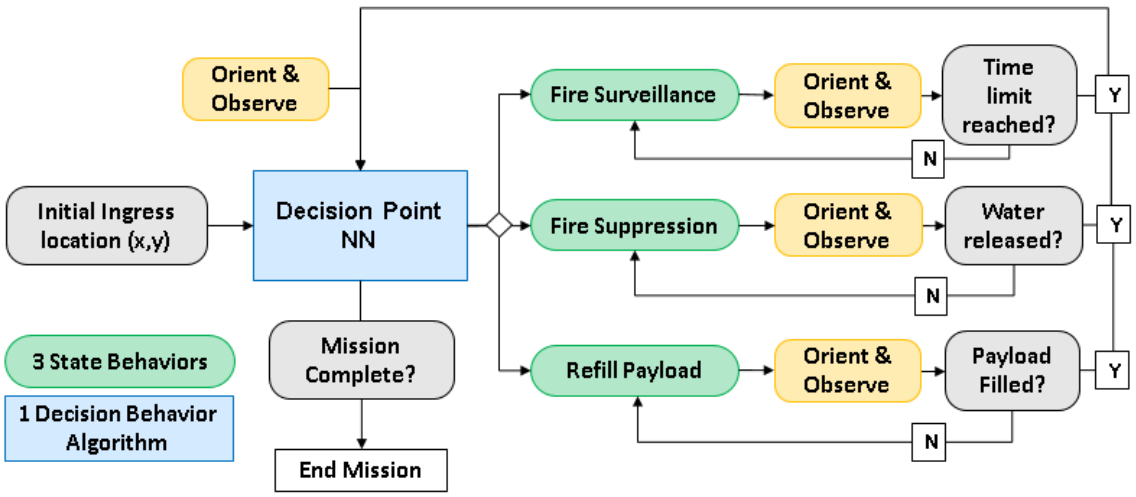

Figure 5 – Wildfire surveillance mission logic.

behavior and each state behavior has a unique set of states, observations, and actions available to the agent.

4.3.1 Behavior Input Definition

The POMDP problem is formulated using continuous state variables, of which the aircraft has perfect knowledge, and an observation image, of which represents the source of imperfect information. Several images define the wildfire's true location, the burnable fuel in the region, and the time since each cell was last visited. These maps are defined in [18]. In addition, a fourth image is required that maps the location of nearby water sources. Each map, illustrated in Figures 6 and 7, is a $100X100$ grid that has normalized values between zero and one.

As the agent explores the region, its knowledge of the fire within a small radius around the agent will update from the truth map to it's observation map. The observation map is the image provided to both the decision and state behavior algorithms. The observation image is rotated to the agent's point of view and centered on its current location, illustrated in Figure 6. The wildfire observation image is updated based on observations within the agent's field of view. The field of view for all state behaviors is arbitrarily set at a 100 meter radius.

The fire surveillance and fire suppression state behaviors receive the rotated observation map of the wildfire as inputs while the payload refill state behavior receives the rotated observation map of the water sources. The knowledge of water source locations is assumed to be known at all times since this information could be determined a priori.

Each state behavior also receives two continuous state variables. The agent's current bank angle, ϕ, and a relative flight path angle, ψ. Surveillance uses the agent's relative angle to the closest burning cell from the true map of the wildfire. Once the wildfire is within view, the relative angle for surveillance changes to the average location of burning cells within view that are in front of the agent, which encourages a smooth change in the angle. Fire suppression uses the agent's relative angle to the estimated drop location and payload refill uses the relative angle to the center of the closest water source. The relative angle informs the agent if its heading towards its respective point of interest. Each relative angle is in the bounds $[-180, 180)$ $degrees$, where a value of zero signifies the agent is heading directly towards the point of interest. The agent uses this information, along with its bank angle, to determine if it should increase or decrease its bank angle. The actions available for all state behaviors are to increase or decrease the agent's bank angle by a rate of $\pm 5 deg/sec$. The agent is unable to choose a $\Delta\phi$ of zero in order to reduce the dimensionality of the state-action mapping [18].

The decision behavior algorithm takes as input both the centered and oriented observation map and the true water source location map. This provides the agent with more knowledge to aid in its decision-making throughout the mission. The continuous state variables provided to the algorithm

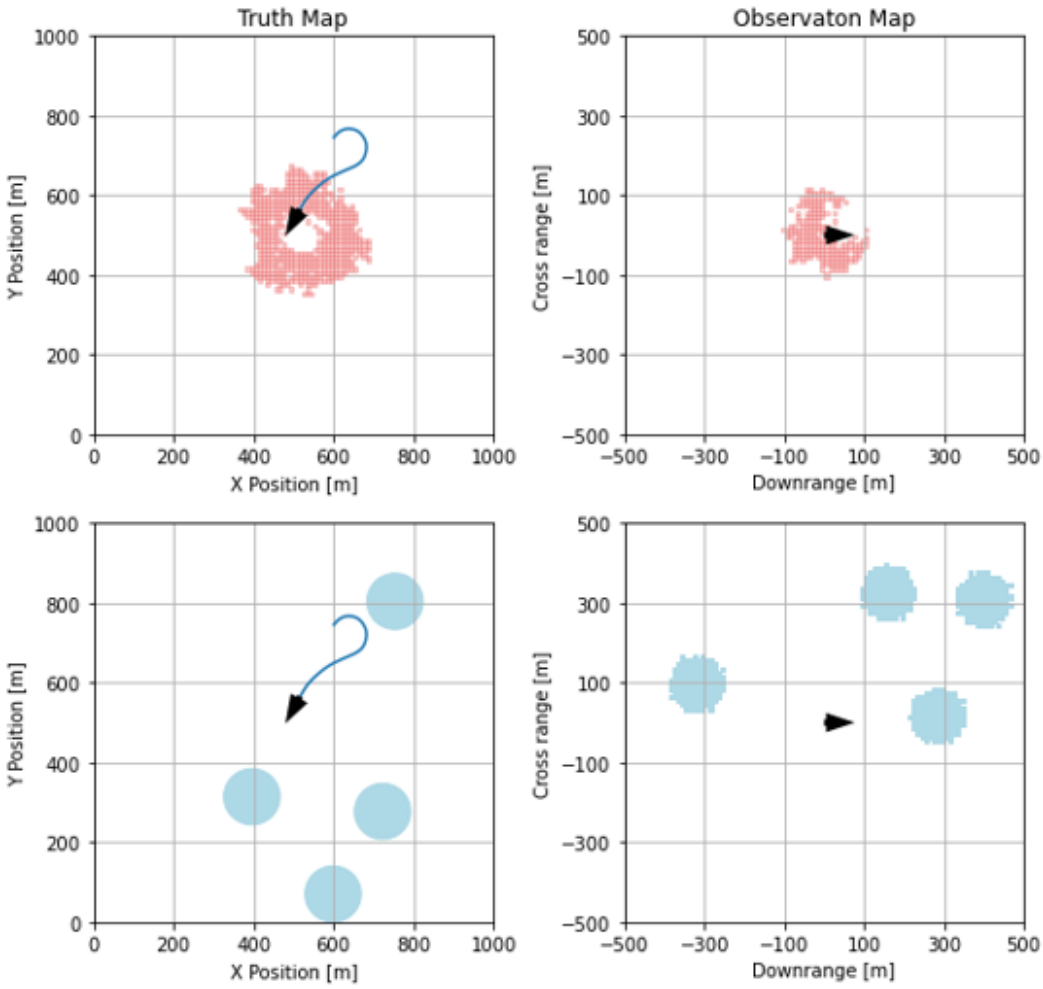

Figure 6 – The true and observed wildfire and water source maps.

include the agent's perceived knowledge and the percent of water remaining within the aircraft. The perceived knowledge is determined by summing the product of the observed wildfire map by the time since each cell was last visited. The observed wildfire map is an array of binary numbers where a value of 1 corresponds to a burning cell and a value of 0 corresponds to a nonburning cell. The time since last visited map is a an array of ones that decreases to zeros over 255 time steps. This formulation results in high values for burning cells that were just observed and low values for burning cells that have not been visited recently. Higher perceived knowledge values indicates higher accuracy of the information being used to make a decision. The actions available to the decision behavior algorithm are the three available state behaviors. Each time an action is chosen, the agent will perform that state behavior and return a corresponding reward.

Each pair of observations and state variables require a unique ANN architecture. In order to incorporate both images and continuous state variables, both a convolutional neural network (CNN) and fully connected network are combined. The CNN contains three sets of a convolutional layer followed by a max polling layer. The convolutional layers have a kernel size of 3, stride length of 1, and zero padding. The max pooling layers have a kernel size of 2 and zero padding. The output of the CNN is flattened and connected to two fully connected layers of 500 and 100 units respectively. The continuous state variables are passed through two fully connected layers of 100 units each and then combined with the output of the CNN. These two neural networks are combined sent through two additional fully connected layers of 200 and 100 units respectively. This network architecture was used for all four behavior algorithms with the exception that the decision behavior algorithm's CNN has two channels, one for each observation map, and an additional output to account for the additional state behavior available to the decision behavior algorithm.

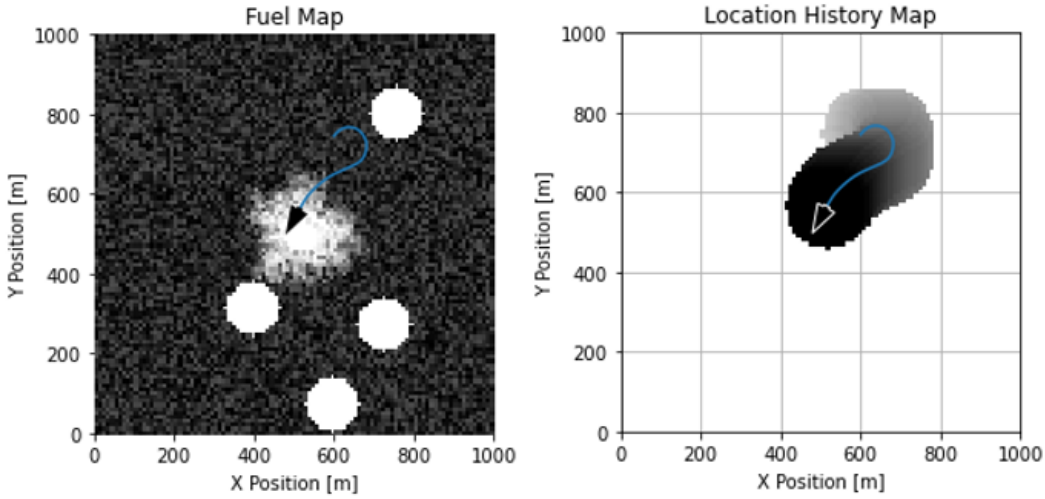

Figure 7 – The fuel map shows the remaining forest and the location history map shows recently observed regions the agent has visited.

4.3.2 Reward Model

Each State behavior has a unique reward or set of rewards during each time step. Fire surveillance receives three rewards given at each time step along with a final reward given for the amount of time spent in view of the fire. The surveillance behavior is rewarded for moving closer to the fire front to encourage the agent to stay near the fire. The distance to the fire front, d, is determined from the set of burning cells, $B(s)$, where cell s is in the set of all cells S. High bank angle is penalized for the surveillance behavior to discourage the agent from flying in tight circles. The third reward is given for the amount of burning cells within the agent's field of view, which encourages the agent to stay over the fire once it is found. The three reward equations for the surveillance state behavior are listed below.

$$R_1 = -\lambda_1 \min_{s \in S | B(s)} d_s \tag{1}$$

$$R_2 = -\lambda_2 \phi^2 \tag{2}$$

$$R_3 = \lambda_3 \sum_{s \in S | d_s < r} B(s) \tag{3}$$

where each λ is a reward tuning parameter, ϕ is the agent's bank angle, and r is the radius of the agent's field of view.. In addition to these rewards, a surveillance state behavior receives an additional reward for the amount of time it spent over the fire. This reward is calculated as the number of time steps where a burning cell was within the agent's field of view.

The reward received during each time step of the refill payload behavior is calculated as the change in distance to the closest water source, d_w. This state behavior calculates the closest water source at the start of its execution and does not update the agent's target water source. The closest water source is only updated once the target water source is reached and the decision behavior algorithm selects the refill state behavior again.

$$R_4 = \lambda_4 \left(-1 + \frac{d_{w,t-1} - d_{w,t}}{v \times dt} \right) \tag{4}$$

The fire suppression state behavior is similar to equation 4, but the point of interest is the agents distance to the planned drop point, d_p. This drop point is where the agent will release its payload to extinguish the fire within a specified radius. The drop point is initially calculated using the agent's observations of the fire and available fuel. A point is picked that maximizes the number of burning cells with the most amount of fuel left to burn. Every 100 time steps, the drop point is shifted 30%

closer to the true optimal drop point. This shift in the drop point accounts for the expected movement of the fire during the time the agent takes to arrive at the drop point.

$$R_5 = \lambda_5 \left(-1 + \frac{d_{p,t-1} - d_{p,t}}{v \times dt} \right) \tag{5}$$

The decision behavior function receives a unique reward at the end of each state behavior along with a final reward once the fire reaches the edge of the map or the maximum number of 25 decisions were made. The selection of the fire surveillance state behavior returns a reward representing the agent's change in knowledge about the wildfire. The change in knowledge is calculated by the difference of two sums. The first sum is the amount of burning cells that are believed to be burning, $B_{obs}(s)$, and actually burning, $B_{true}(s)$, at initial time step, t_0, while the second summation is the same, but calculated at the final time step, t. The fire suppression state behavior will return a reward proportional to the amount of burning cells extinguished. The fire surrounding the drop point is extinguished based on a normal probability density function that results in a higher probability of extinguishing the fire closer to the agent. Finally, the selection of the refill state behavior returns the summation of one minus the normalized time taken to reach a water source and the percent of payload refilled. These rewards are stated below.

$$R_6 = \lambda_6 \left(\sum_{s \in S | B_{t_0,obs}(s), B_{t_0,true}(s)} B_{t_0}(s) - \sum_{s \in S | B_{t,obs}(s), B_{t,true}(s)} B_t(s) \right) \tag{6}$$

$$R_7 = \lambda_7 \left(\sum_{s \in S | B(s)} B(s) \right) \tag{7}$$

$$R_8 = \lambda_8 \left[(1 - \triangle t) + \left(P_{filled} / P_{max} \right) \right] \tag{8}$$

where $\triangle t$ is the length of the state behavior's time interval, and P_{filled} and P_{max} are the amount of water refilled and the maximum payload capacity. The maximum payload capacity was set at two, which corresponds to the agent dropping water twice at the end of two separate fire suppression state behaviors. The final reward the decision behavior algorithm receives once the mission completion criteria is triggered is the summation of the agent's knowledge of the wildfire at each decision point. This reward encourages the suppression of fire by reducing the wildfire's size and the surveillance of the fire by increasing the agent's observations.

4.4 Algorithm Definition and Training

Many RL algorithms exist that could be suitable for the proposed decision behavior approach. Implementation of this proof-of-concept utilizes Proximal Policy Optimization (PPO) due to the successes of policy search methods with POMDPs [20]. PPO is a recently introduced family of policy gradient methods for RL. These methods build on knowledge gained from trust region policy optimization to develop methods that maintain their stability and reliability while providing better performance with a much simpler implementation [21]. Two key aspects of PPO are its clipped surrogate objective function and its policy update method, which performs stochastic gradient ascent over multiple epochs. The PPO Algorithm used in this study was modified from [22].

4.4.1 State Behavior Training

Once the algorithm is outlined, tuning parameters must be defined and explored for each state behavior to improve the algorithm's final policy. Hyperparameter tuning for each state behavior can be performed independently from one another due to their non-conflicting mission tasks. The final set of parameters used were unique for each state behavior due to each behavior's variation in mission duration. Fire surveillance suppression used a discount factor of 0.99 while payload refill used a discount factor of 0.85. This factor was used to determine the discounted reward along an entire mission envelope for each behavior. All three behaviors had a learning rate of 0.001. In addition to these parameters, a batch size of 100 was used for each state behavior and the number of policy update

epochs was set at three. Each behavior is trained for 5000 episodes to produce a satisfactory behavior. The convergence plots for each behavior are illustrated in Figure 8. The increase in average rewards, denoted by the orange line, for both the total episode rewards and mean batch rewards illustrates that the algorithms converge towards an optimal solution. Each behavior was trained for 1–3 hours on a single GPU.

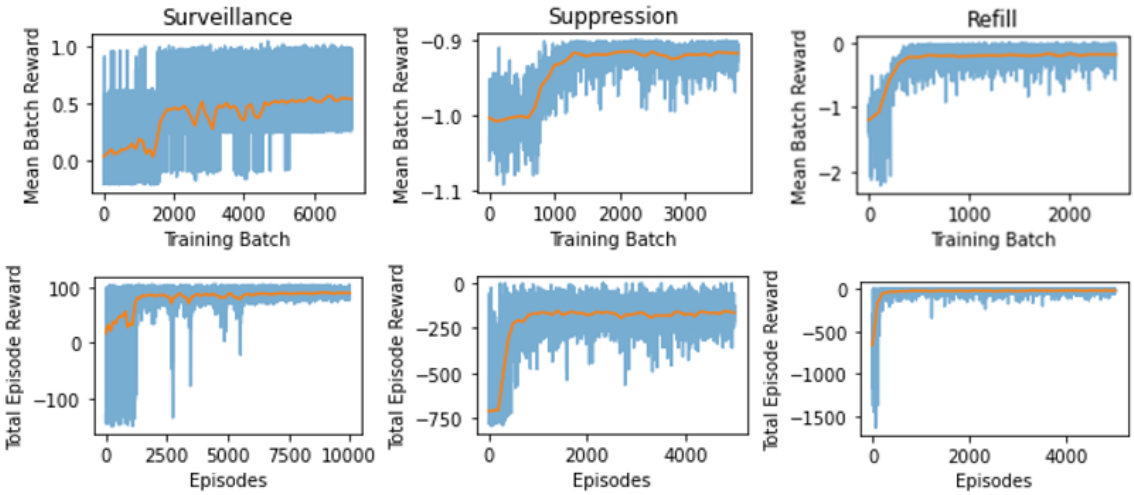

Figure 8 – Mean batch rewards and total episode rewards during algorithm training for each state behavior. The orange line denotes the average, calculated from intervals of 100.

The resultant surveillance, suppression, and refill behaviors are illustrated in Figure 9, respectively. The fire surveillance behavior successfully finds the fire front and observes its movement for 100 time steps for all 200 test episodes. The grey area surrounding the agent's path highlights the agent's field of view that is used to update the observation map. The fire suppression behavior, shown in the middle, also successfully reaches its drop point for all test episodes. The drop point, highlighted in black, can be seen slowly shifting down as the true fire spreads. The final refill behavior is shown on the right. Similar to the other state behaviors, the agent successfully achieved its goal of flying to the closest water source every test episode.

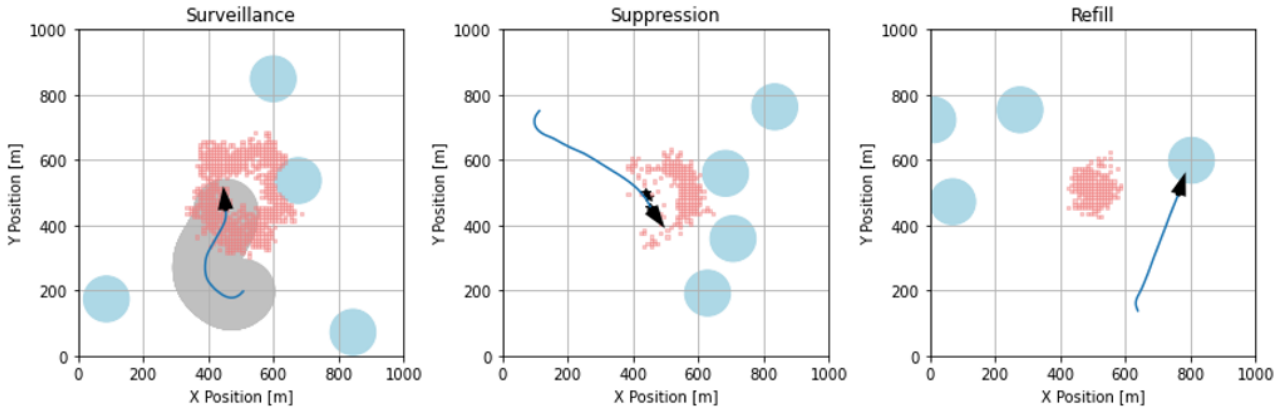

Figure 9 – Trained state behaviors performing a mission segment.

4.4.2 Decision Behavior Training

Similar to the state behavior algorithms, the decision behavior algorithm requires tuning of the hyperparameters. A learning rate of 0.001 and a discount factor of 0.99 was used along with a batch size and policy update epoch of 32 and three respectively. The decision behavior algorithm was trained for 3000 episodes. Convergence of the decision behavior algorithm is illustrated in Figure 10. Both

the total episode reward and mean batch reward converge towards an optimal solution, but further hyperparameter tuning may improve convergence.

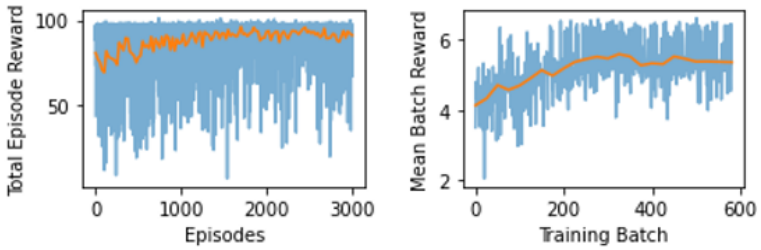

Figure 10 – Mean batch reward and total episode reward during algorithm training for the decision behavior. The orange line denotes the average, calculated from intervals of 25.

Figure 11 illustrates the progression of the decision behavior algorithm as the wildfire spreads along with the agent's corresponding observation images of where it has observed the wildfire to be burning. The agent initially starts at $x = 500$ and $y = 100$, denoted by the green dot, with initial flight path and bank angles of zero. The initial location of all burning cells at $T = 0$ is known. Each row of plots illustrates a progression of five decisions. The end of each state behavior is denoted by a star which is colored respectively to its legend entry.

The agent enters the impacted area without any initial payload and chooses to immediately fill its payload from the nearby water source as its decision. Once its payload is filled, the agent chooses to conduct two fire suppression behaviors to attempt to immediately contain the fire. Once its payload is depleted, the agent chooses to conduct fire suppression. At this point, the agent has a low perceived knowledge of 35%. A perceived knowledge of 100% corresponds to perfect knowledge of the wildfire. The agent then chooses two consecutive surveillance behaviors to raise its perceived knowledge to 80%. Once the agent raises its perceived knowledge, it chooses to conduct fire suppression as its fifth state behavior of the mission. This progression of the first five state behaviors is only one example of a possible set of decisions the agent could make during wildfire surveillance and management.

4.5 Simulation Results

The decision behavior algorithm is sampled for 200 missions each with the same initial conditions. Variation in outcome is due to the stochasticity of the PPO algorithm. The frequency of each state behavior with respect to the state variable is illustrated in Figure 12. Fire surveillance is chosen more frequently for lower perceived knowledge; however, all three state behaviors were found to be skewed towards a lower perceived knowledge. This finding is due to the lower sampling frequency of higher perceived knowledge values due to the rapid spread of the wildfire decreasing the agent's knowledge. Revising the calculation of the perceived knowledge state variable may lead to improved results. The fire suppression state behavior is chosen more frequently as its payload approaches full capacity. For this state behavior, 50% payload capacity corresponds to the minimum amount needed for effective fire suppression. The payload refill behavior is chosen more frequently as its payload capacity decreases. In contrast, decision behavior algorithms preference to choose the fire surveillance state behavior seemed to not be impacted by payload capacity.

The MoEs chosen for the wildfire surveillance and suppression mission are the agent's total time over the fire, including time during all three state behaviors, and the amount of burning cells extinguished by the agent. These MoEs are normalized by their highest value sampled, respectively. Figure 13 illustrates the comparison of MoEs for each sampled mission solution. The optimal operational effectiveness is the set of points that are not dominated by others in either the x or y direction. These points are known as the Pareto frontier where each point in the set represents a set of sequential decisions – or tactic – the agent used. The variation in operational effectiveness illustrates the exploration of the mission action design space for a single aircraft.

The blue point in Figure 13 highlights the solution that resulted in the highest summation of the normalized MoEs. The first half of this solution is illustrated by Figure 11. The final set of 20 decisions the agent makes for this solution is illustrated in Figure 14. Although the optimality of a single solution

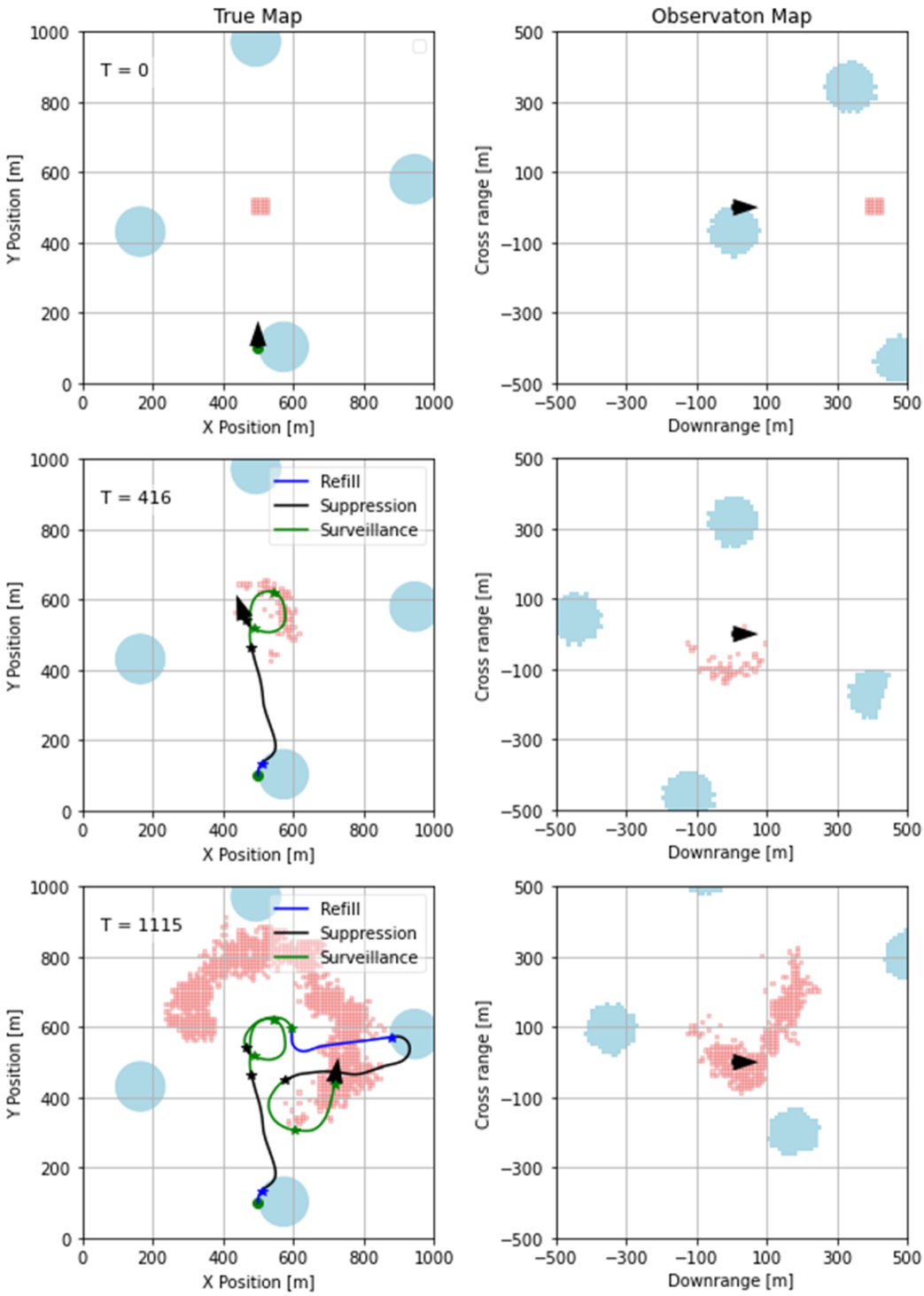

Figure 11 – Agent traversing the mission using the decision behavior approach along with its corresponding observation map of the wildfire.

depends on the importance the decision-maker places on each MoE, this point is still considered optimal since it is contained in the set of points that make up the Pareto frontier.

One behavior to note from the highlighted solution is that the agent seems to focus on the bottom left fire front. This behavior could be due to the rapid speed of the fire spreading, which results in poor solutions that waste valuable time crossing over the map to monitor multiple sides of the fire front. Another noticeable trend is the lack of surveillance behavior in the second half of the mission. This could be due to the increased travel time for the suppression and refill behaviors, which continually update the observation map even though the agent is not in its surveillance behavior.

These results highlight the ability to quantify the operational effectiveness for a minimally defined

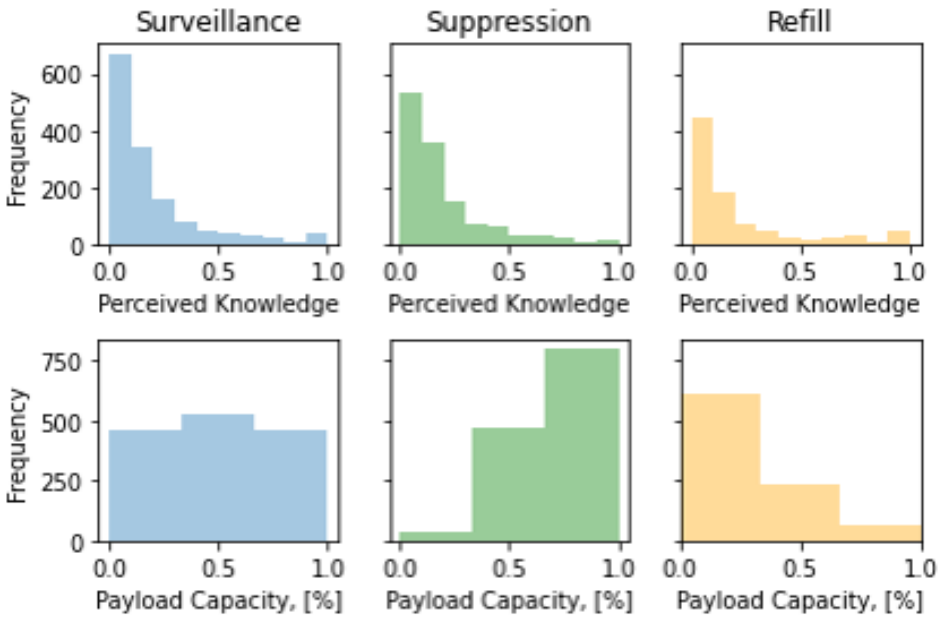

Figure 12 – Frequency of state behavior selection for each continuous state variable input into the neural network.

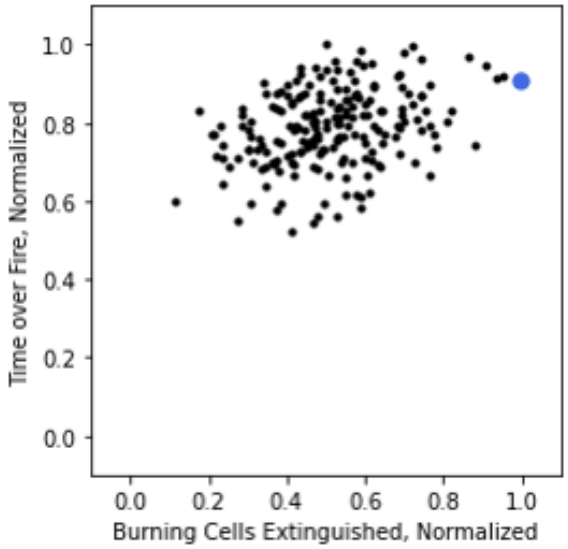

Figure 13 – Comparison of the MoEs for each sampled mission.

mission using the decision behavior approach. The traceable set of decisions made for each solution provides valuable information about the collective mission behavior.

5. Conclusion

The proposed agent-based decision behavior approach seeks to enable the exploration of the ways, or more formally, the mission action design space. The proof-of-concept wildfire surveillance and suppression mission successfully demonstrated the decision behavior approach. The discretization of the mission into decision and state behaviors enable the quantification of operational effectiveness for a minimally defined mission. This approach provides traceability of the collective mission behavior and addresses the dimensionality of the mission action design space through critical decision points. The quantification of the aircraft's operational effectiveness provides additional information for the decision-maker during technology investment.

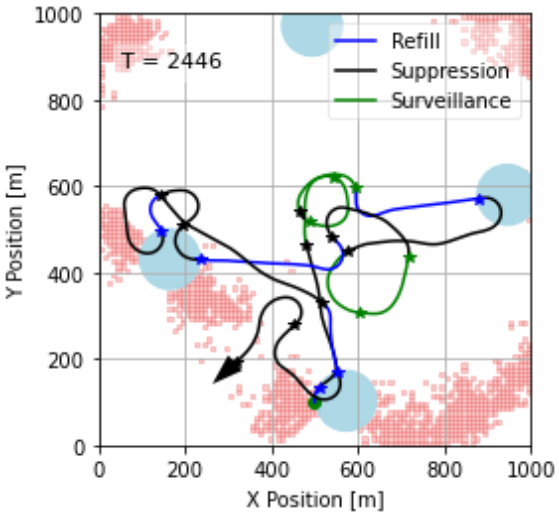

Figure 14 – One solution from the set of optimal solutions of the decision behavior algorithm.

6. Contact Author Email Address

mailto: ahull7@gatech.edu

7. Acknowledgements

The authors would like to thank Dr. Michael Steffens for his role as a technical advisor throughout the development of the technical approach.

8. Copyright Statement

References

[1] Federal Firefighting Costs (Supression Only). *National Interagency Fire Center*, 2020. Retrieved May 2021.

[2] Butler C, O'Connor m AND Lincoln j. Aviation-Related Wildland Firefighter Fatalities – United States, 2000–2013. *Morbidity and Mortality Weekly Report*, Vol. 26, No. 29, pp 793–796, 2015.

[3] Ambrosia V, Wegener S, Zajkowski T, Sullivan D, Buechel S, Enomoto F, Lobitz B, Johan S, Brass J and Hinkley E. The Ikhana unmanned airborne system (UAS) western states fire imaging missions: from concept to reality (2006–2010). *Geocarto International*, Vol. 26, No. 2, pp 85–101, 2001.

[4] Mavris D and Sudol A. Formulation and implementation of a method for technology evaluation of systems of systems. *31st Conference of the International Council of the Aeronautical Sciences*, Belo Horizonte, Brazil, 2018.

[5] Allison D and Kolonay R. Expanded MDO for effectiveness based design technologies: EXPEDITE program introduction. *2018 Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, USA, 2018.

[6] Reuter R, Iden S, Snyder R and Allison D. An overview of the optimized integrated multidisciplinary systems program. *57th AIAA / ASCE / AHS / ASC Structures, Structural Dynamics, and Materials Conference*, San Diego, CA, USA, 2016.

[7] Gordon, S. A Stochastic Agent Approach (SAA) for mission effectiveness. Georgia Institute of Technology, Atlanta, GA, USA, 2018.

[8] Siebers P, Macal C, Garnett J, Buxton D and Pidd M. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, Vol. 4, No. 3, pp 204–210, 2010.

[9] Macal C and North M. Introductory tutorial: agent-based modeling and simulation. *the IEEE Winter Simulation Conference*, pp 6–20, 2014.

[10] Sakellariou I. Agent based modelling and simulation using state machines. *Second International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, Fordham University, NY, USA, 2012.

[11] Volovoi V. Abridged Petri Nets. *arXiv preprint arXiv:1312.2865*, 2013.

[12] Banisch S. *Markov Chain Aggregation for Agent-Based Models*. Springer International Publishing, 2016.

[13] Sutton R and Barto A. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[14] Haas P and Shedler G. Stochastic petri net representation of discrete event simulations. *IEEE Transactions on Software Engineering*, Vol. 15, No. 4, pp 381–393, 1989.

[15] Åström K. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, Vol. 10, No. 1, pp 174–205, 1965.

[16] Kaelbling L, Littman M and Cassandra A. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, Vol. 101, No. 1, pp 99–134, 1998.

[17] Kaelbling L, Littman M and Moore A. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, Vol. 15, No. 4, pp 237–285, 1996.

[18] Julian K and Kochenderfer M. Distributed wildfire surveillance with autonomous aircraft using deep reinforcement Learning. *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019.

[19] Daberkow D and Mavris D. An investigation of metamodeling techniques for complex systems design. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, USA, 2002.

[20] Singh S, Jaakkola T and Jordan M. Learning without state-estimation in partially observable Markovian decision processes. *International Conference on Machine Learning (ICML)*, 1994.

[21] Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[22] Barhate N. Minimal PyTorch Implementation of Proximal Policy Optimization. *GitHub repository*, 2021.