

A Distributed Deep Learning Approach for A Team of Unmanned Aerial Vehicles for Wildfire Tracking and Coverage

Kripash Shrestha, Hung Manh La, *IEEE Senior Member*, Hyung-Jin Yoon

Abstract—Recent large wildfires in the United States and the subsequent damage that they have caused have increased the importance of wildfire monitoring and tracking. However, human monitoring on the ground or in the air may be too dangerous and therefore, there need to be alternatives to monitoring wildfires. Unmanned Aerial Vehicles (UAVs) have been previously used in this problem domain to track and monitor wildfires with approaches such as artificial potential fields and reinforcement learning. Our work aims to look at a team of UAVs, in a distributed approach, over an area to maximize the sensor coverage in dynamic wildfire environments. We proposed and implemented the Deep Q-Network (DQN) with a state estimator (auto-encoder), then compared it to existing methods including a Q-learning, a Q-learning with experience replay, and a DQN. The proposed DQN with a state estimator outperformed existing deep learning methods in terms of reward maximization and convergence.

I. INTRODUCTION

Wildfires are spontaneous events that are known to cause massive destruction to structures and wildlife. Consequently, fighting wildfires is a dangerous task as the behavior of wildfires can be unpredictable and difficult to model. The Congressional Research Service reported that for every year since 2000, there have been an average of 70,600 wildfires that burned an average of 7.0 million acres in the United States alone [1]. Although on average, there were 78,600 wildfires annually in the US in the 1990s, the total acres burned have more than doubled. The loss of wildlife, human life, and structures highlights the importance of the need to locate, observe and track wildfires. This information is critical to making emergency plans to evacuate civilians to safety, prevent the rapid spread of the fire and to fight the fire.

UAVs have been used to assist humans in emergency and disaster situations by providing situational awareness with information such as imagery and maps [2]–[6]. By maintaining proper communication links between the UAVs and human operated ground control stations, humans can remotely and safely assess damage in a given region of interest and the activity of such disasters [7]–[9]. Therefore, UAVs are highly suitable for tackling the wildfire tracking problem by providing imagery and maps while relaying information through each other to firefighters/operators who are at a safe distance. Teaming of UAVs and other robots to collaborate for resolving multiple challenges has risen in popularity for both research

and application [10], [11]. Teams of UAVs have been used to collaborate as a network of agents with sensors to build maps and gather information in their local areas [12], [13].

The objective of this work is to use a team of UAVs to track the spread of a wildfire and monitor the entire fire. This work takes inspiration from [14], [15], which uses UAVs and multi agent reinforcement learning to control UAVs for static field coverage. Previous work has also been done [16], [17] that formulated fire boundary tracking as a two-objective optimization problem, using the same artificial potential fields-based approach as described in [17] to control these UAVs. Specifically, Shrestha et al. [18] maximizes the fire boundary coverage and minimizes the energy consumption of deployed UAVs in a network. The first objective ensures that agents could track the spread of the fire while the second maximizes the length of time UAVs, which can stay in the air while tracking the fire. The work uses its own implementation of Deb's NSGA-II [19] to optimize potential field parameters encoded in a real-valued chromosome. The tuned potential field parameters control a team of UAVs that maneuver autonomously to track the spread of the fire front in a distributed manner. Past work has shown that potential fields are suitable for distributed control tasks since, once tuned, they have low computational requirements, can be visualized to help explain control behavior, and have been used to produce complex task achieving behavior for heterogeneous groups of agents [16], [17], [20], [21]. However, potential fields can be highly non-linear and that makes it difficult to optimize.

In this paper, we are interested in finding alternative and other methods to track the entire coverage of the fire with a dynamic fire environment. Previous works done have mostly prioritized just the fire fronts with a dynamic fire environment or just total fire coverage with a static fire environment. We will implement and compare Q-Learning, experience replay with Q-learning, Deep Q-Network (DQN), DQN with a state estimator (auto encoder) to track and cover wildfires

The remainder of the paper is organized as follows. Section II describes prior work in using UAVs for wildfire tracking. Section III briefly describes our problem formulation [14]. Section IV discusses the different methods that were used for the problem [22]. Section V presents experimental results and compares the results with each other. Finally, Section VI summarizes the work and improvements for future work.

II. BACKGROUND

Although UAVs, a field of multi-robot systems [23]–[26], have many applications in both civilian and military domains [27]–[30], we focus on using UAVs for tracking and covering wildfires. Two fundamental challenges when using UAVs in unknown regions for fire coverage are fire detection and fire tracking. However, the two are related as we need detection for tracking, so we start with providing an overview of work in

Kripash Shrestha is with Amazon Robotics Research. Hung La is with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, USA. Hyung-Jin Yoon is with the Department of Mechanical Engineering, University of Nevada, Reno, USA. Corresponding author: Hung La, email: hla@unr.edu.

This work is supported by the Vingroup Joint Stock Company and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.NCUD.DA094. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the VINIF.

detection. Yuan et al. [31] used Infrared (IR) imaging sensors installed on UAVs to detect the presence of fire and presented techniques to process images gathered using different sensors mounted on UAVs, to study fire spreading behaviors. Afghah et al. [32] proposed a leader-follower formation to cluster a set of UAVs into multiple coalitions that collectively covered a particular area of interest. Merino et al. [33] proposed a cooperative perception system for multiple heterogeneous UAVs for automatic detection of forest fires. They collected data using multiple sensors such as visual cameras, infrared sensors, and fire detectors mounted on UAV's and fused them together for detection, monitoring, and measurement of forest fires. Another fire detection technique developed by Yuan et al. [34] analyzes fire segmentation in different color spaces. Cruz et al. [35] introduced a Forest Fire Detection Index (FFDI) to detect fires through the use of a new color index. The index is based on method for vegetation classification and used to detect flames and smoke.

The work in fire detection has shown that several robust techniques exist for fire detection and tracking as long as we have good observation platforms with suitable sensors. In this paper, we assume that UAVs have a camera that is able to observe the fire, and the challenge is to cover the fire entirely as it spreads over time. Coordinated control of multiple UAVs is essential for dynamic fire coverage, and control techniques are broadly categorized as either distributed or centralized control. Casbeer et al. [36] presented a path planning algorithm to track fire using low altitude short endurance UAVs. Their centralized path planning computes waypoints for each UAV, with these waypoints being generated along the edge of a fire (along the fire boundary). Phan et al. [37] worked on a similar problem where they proposed a cooperative control framework for a team of UAVs and unmanned ground vehicles (UGVs) to detect and track fires. In their centralized framework, a mission controller monitors a dynamic environment, formulates high level mission plans, and allocates task to each vehicle.

Maza et al. [38] proposed a distributed decision making architectural framework for multi-UAV configuration in disaster management. Multiple checkpoints in the region of interest can be used to command UAVs to track down the boundary for fire coverage. Kyrkou and Theocharides [39] presented a deep learning approach to deploy UAVs to collect images for fire classification.

Open challenges still include the coverage of a fire when the fire spreads dynamically. Earlier approaches to resolve this problem include [36] and [37] where UAV maneuvering decision making was centralized. Centralized approaches suffer from a single point of failure and may need more computation and communication hardware than available. However, given enough computational and communication resources, these approaches can direct a team of UAVs to continuously track the spread of fire along the boundary as long as we have a good fire spread model or external information about fire boundary. Simultaneous detection and tracking in a decentralized manner remains challenging. Pham et al. [16] used a team of UAVs working together to track and follow a wildfire as it spreads by tracking fire intensity and heat sources. However, an artificial potential field-based method was used and required time to tune its parameters as they can be highly non-linear and that makes them difficult to optimize.

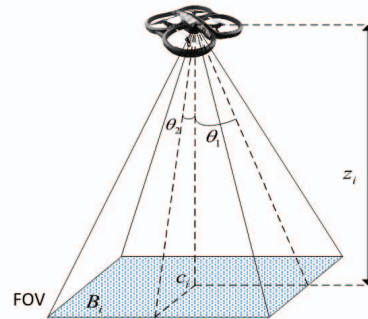


Fig. 1: Field of view of a UAV with half angles θ_1, θ_2 . The center of the camera is located at the UAV's lateral coordinate at c_i . Each UAV will record/capture an image of its observations using its camera and record the number of fire grids it observes.

III. PROBLEM STATEMENT

In this work, we aim to deploy and control a team of UAVs to track and cover a wildfire in unknown environments. Let us assume that we have a set of n homogeneous UAVs. We assume that the UAVs are equipped with a GPS receiver, a camera capable of detecting the fire, and able to move North, East, West, South, and change elevations. We assume that the UAVs also start at known fire locations at deployment but has to learn to track and cover the wildfire as it spreads and moves. At initial deployment, each UAV starts at a fire location that is not occupied by any other UAV. The coverage and tracking objective consists of the UAV team spreading out and learning to cover the fire as it spreads. Each UAV will capture images of the area under its observation with the camera. The camera's FOV model is a pyramid and based on its half-angles $\theta^T = [\theta_1, \theta_2]^T$. Additionally, we make the assumption that the UAVs are able to communicate to each other and can share their positions with each other at each time step.

For experimentation and simulation, the fire model is simulated using FARSITE [40] fire model. Different FARSITE parameters were used to simulate and obtain different behaviors of various wildfire behavior. The fire model and spread are represented in a discretized grid environment. The objective of the team of UAVs is to cover the fire as much as possible, and this is represented by the reward function. Let $p_i = [c_i, z_i] = [x_i, y_i, z_i]$ denote the pose of a given UAV $i \in n$. c_i represents the coordinates in x_i and y_i , the lateral coordinates and z_i represents the altitude of a UAV i . The camera and FOV of the UAVs are identical to [14] and shown in Figure 1 below.

For deployment, the UAVs are initially deployed at a distinct fire location. The UAVs start at the known fire locations in the discretized grid space. After the initial deployment, the UAVs now move and learn to cover the fire as a team as the fire spreads. The problem has been solved using potential fields [16], [18] and Voronoi partitions but potential fields can be highly non linear and difficult to tune. Additionally, some of the previous works make assumptions about the mathematical model of the environment. However, this information is difficult to model accurately because there is insufficient information or the constant change in the model because of the dynamic environment.

IV. METHODOLOGY

A. Simulation

The fire scenarios in this work is simulated using the FARSITE fire model with different parameters. The parameters are defined as:

- Wind Direction (θ), $0 \leq \theta \leq 2\pi$
- Wind Speed (U), 5 m/sec
- Fire Spread Rate (R), 25 m/min .

To implement the reward function in a simplistic and achievable manner, the environment is implemented into a discretized grid-based environment to describe the fire model. The state space and action space set of each UAV is represented as a discrete finite set in a 3-D grid environment.

B. Reinforcement Learning

Reinforcement Learning aims to maximize the rewards that a UAV receives once it interacts with the environment. The UAV takes an action at each state based on a policy π to maximize the reward function, which measures the performance of the UAV. However, in this problem there are various UAVs that interact with each other and the environment. This makes the interaction more complex as the states and environment are constantly changing. In our approach, we form a joint action containing all actions of all UAVs taken at a single time step. This allows each UAV to consider the states and actions of all other UAVs to coordinate for the optimal state-action pairs. We define the learning model as:

- n is the number of UAVs.
- S is the joint state space $S = S_1 \times S_2 \times \dots \times S_n$. S_i , where $i = 1, \dots, n$, represents the individual state space of a UAV i .
- A is the joint action space $A = A_1 \times A_2 \times \dots \times A_n$. S_i , where $i = 1, \dots, n$, represents the individual action space of a UAV i .
- T is the transition probability function, $T : S \times A \times S \rightarrow [0, 1]$.
- R is the reward function, $R : S \times A \rightarrow \mathbb{R}$.

The UAVs aim to optimize their reward output by determining the action to take that will return the highest reward in the future. Since the problem contains multiple UAVs, the expected return obtained over an episode depends on the joint state and action pairs. In the multi-UAV scenario at time t , the Q-learning function can be represented as $Q(S_t, A_t) = Q(S_{i,t}, A_{i,t}, S_{i,t-1}, A_{i,t-1}) = E \sum_{n=t}^{\infty} \lambda r_{i,t+1}$. λ , defined by, $0 \leq \lambda \leq 1$ is the discount factor of Q-learning and effects to prioritize the rewards received earlier more or prioritize the rewards received later more. A value closer to 0 will make the UAV consider earlier/current rewards, whereas a value closer to 1 will prioritize the rewards further into the future. In [14], the authors proposed Correlated Equilibrium to accomplish the team's goals by reaching a consensus in selecting the actions. We consider their method of Correlated equilibrium for helping find a joint action for two of the methods implemented for comparison. We implemented Pham et al.'s work [14] as a baseline comparison for the dynamic wildfire situation, but also implement Experience replay to Pham et al.'s work, a Deep Q-Network (DQN) approach, and a DQN with state estimator approach for comparison.

C. Reward and Q-Learning Design

[14] showed that a global team goal results in better performance and rewards than individual rewards as it converged faster and resulted in higher reward. The global reward is defined as the total number of fire cells observed at a time t over the total fire cells that could be observed at a time t . Therefore, the global reward function is simply defined as the percentage of the fire observed by the team of UAVs. The global reward equation can be represented as: $GR = \frac{\sum_i f_i(S_t)}{FC_t}$ where f_i is the number of fire cells observed by UAV i at time t , and FC is the total number of fire cells observable at time t . We implement the individual reward for each UAV to be equal to that of the global reward equation as the UAV's share the same team objective and goal. The Q learning function can then be approximated as:

$$Q^{new}(S_t, A_t) = (1 - \alpha)Q(S_t, A_t) + \alpha(GR(S_t, A_t) + \lambda \max_{A' \in A_t} Q(S_{t+1}, A')),$$

where, α , defined by, $0 < \alpha < 1$ is the learning rate, which determines how quickly the learning is updated. A value closer to 0 indicates that the Q-values are not updated as often, but a value closer to 1 indicates that the Q-values are updated often. The Q-learning and Q-learning with experience replay methods proposed in this work use the same fixed sparse representation (FSR) scheme as implemented by [14]. The Q-Learning algorithm is defined in [14].

D. Q-Learning with Experience Replay

Additionally, Experience Replay (ER) is combined with [14] for learning a value function from a sample of experience rather than the most recent state. In ER, the experiences are stored and used later, whereas in standard Q-learning, the most recent experience is used to update the value function. This allows for experiences to be using important experience repeatedly to update and learn the function. The Q-Learning with ER algorithm can be defined as Algorithm 1:

E. DQN

We consider a DQN to replace the original Q-learning approach for the problem. Considering that we have n UAVs and n DQNs (1 DQN for each UAV). We consider the DQN to have an input of the states and actions of the UAVs, which contains the size for the real values that represent the state values for each UAV in the discretized space (x, y, z) , the action taken by each UAV and the global reward associated with each UAV, and an output of n , which represents the actions that each UAV takes. The loss criterion selected for the approach is a Huber Loss function that is less sensitive to outliers in the data than the squared error loss function. The DQN approach is defined as Algorithm 2:

F. DQN with State Estimator

1) *Online Parameter Estimation*: The recursive Hidden Markov model (HMM) estimation [41], [42], which uses a stochastic approximation [43] can be used to estimate parameters of the model. However, the estimation method [41] becomes computationally intractable as we increase the size of the grid map. Hence, we employ the computational tools developed in deep learning that scales with a larger grid map when it uses GPUs. We approximate the Bayesian recursive updates [41] for the state estimation with the dynamic auto-

Algorithm 1: Q-Learning with Experience Replay

Input : discount factor λ , learning rate α , number of experiences to replay K , number of replays N , number of episodes e , number of UAVs n , epsilon policy switching ϵ

```
1  $Q \leftarrow Q_0$ 
2  $L \leftarrow \emptyset$ 
3 for  $episode = 1, 2, \dots, e$  do
4   Randomly initialize state  $S_{i,0}$ 
5   for  $t = 1, 2, \dots$  do
6     for  $k = 0, 1, \dots, n$  do
7       Exchange information with other UAVs to
8       obtain their positions and state
9        $\pi(A_t) =$ 
10      { Optimal joint-action with probability  $1 - \epsilon$ 
11        Random joint-action otherwise
12      }
13      Decide on unique joint action  $A_t$  and take
14      individual joint actions
15      Exchange information about new UAV states
16      Observe global reward  $GR$  at time  $t$ ,  $GR_t$ 
17       $E \leftarrow (S_t, A_t, GR_t, S_{t+1})$ 
18      Store experience into replay buffer  $L$ ,
19       $L \leftarrow (L, E, N)$ 
20      Retrieve  $K$  replays from replay buffer  $L$ 
21      Update for each UAV:
22      _  $Q \leftarrow QFunction(Q, M, K, \lambda, \alpha)$ 
23
24 return(Parameter Vector  $\theta$  for each UAV  $i$  and policy  $\pi$ )
```

Algorithm 2: DQN

Input : number of episodes e , number of UAVs n , epsilon policy switching ϵ

```
1  $Q \leftarrow Q_0$ 
2  $DQN \leftarrow Initialize$ 
3 for  $episode = 1, 2, \dots, e$  do
4   Randomly initialize state  $S_{i,0}$ 
5   for  $t = 1, 2, \dots$  do
6     for  $k = 0, 1, \dots, n$  do
7       Exchange information with other UAVs to
8       obtain their positions and state
9        $\pi(A_t) =$ 
10      { Optimal joint-action with probability  $1 - \epsilon$ 
11        Random joint-action otherwise
12      }
13      Decide on unique joint action  $A_t$  and take
14      individual joint actions
15      Exchange information about new UAV states
16      Observe global reward  $GR$  at time  $t$ ,  $GR_t$ 
17      Update DQN for each UAV:
18      _  $DQN(S_t, A_t, GR_t, S_{t+1})$ 
```

Algorithm 3: DQN with State Estimator

Input : number of episodes e , number of UAVs n , epsilon policy switching ϵ

```
1  $Q \leftarrow Q_0$ 
2  $DQN \leftarrow Initialize$ 
3 for  $episode = 1, 2, \dots, e$  do
4   Randomly initialize state  $S_{i,0}$ 
5   for  $t = 1, 2, \dots$  do
6     for  $k = 0, 1, \dots, n$  do
7       Exchange information with other UAVs to
8       obtain their positions and state
9        $\pi(A_t) =$ 
10      { Optimal joint-action with probability  $1 - \epsilon$ 
11        Random joint-action otherwise
12      }
13      Decide on unique joint action  $A_t$  and take
14      individual joint actions
15      Exchange information about new UAV states
16      Observe global reward  $GR$  at time  $t$ ,  $GR_t$ 
17      State Estimator Observations
18       $SE\_OBS(S_{t+1}, A_t, R_t)$ 
19      State Estimator Step Function  $SE(S_{t+1}, A_t)$ 
20      Update DQN for each UAV:
21       $DQN(S_t, A_t, GR_t, S_{t+1})$ 
22      Update SE for each UAV:
23      _  $SE(O_t, S_t, A_t)$ 
```

encoder network consisting of an encoder¹, a recurrent neural network (RNN)², and a decoder³, as illustrated in Fig. 2.

The deep neural networks in Fig. 2 are trained to predict the future observation in the next time step given previous observations. In contrast to the HMM estimation methods [41], which only use the current observation, we use replay buffer [47] to save recent state transition reward associations to sample mini-batch samples for training. The trajectory of the observation grid map with $|\mathcal{O}|$ possible observations can be seen as $|\mathcal{O}|$ channel image stream. The model in Fig. 2 is trained to predict the observation image streams, i.e. observation grid map trajectories.

Additionally, we consider an online state estimator discussed above with our DQN approach to estimate the state at the next time step. Our DQN algorithm remains the same but now we incorporate a state estimator. The state estimator can be represented as a dynamic auto-encoder that is trained to predict the next state using batch observations from the UAVs, a batch of states of the UAVs, and a batch of actions taken by UAVs. The input for the state estimator is the observation of all UAVs at time t (O_t), the state vector for all UAVs at time t (S_t), and the action taken by all UAVs at time t (A_t). The state estimator is then trained to predict the future observation in the next time step given the previous observations. The DQN with State Estimator algorithm can be defined as Algorithm 3:

We can represent the methodology as shown in Fig. 3. The

¹The encoder compresses images into encodings [44].

²We use the gated recurrent unit (GRU) for the RNN component [45].

³The decoder generates a grid map (an M channel image). The generator employs the deconvolutional layer structure in [46].

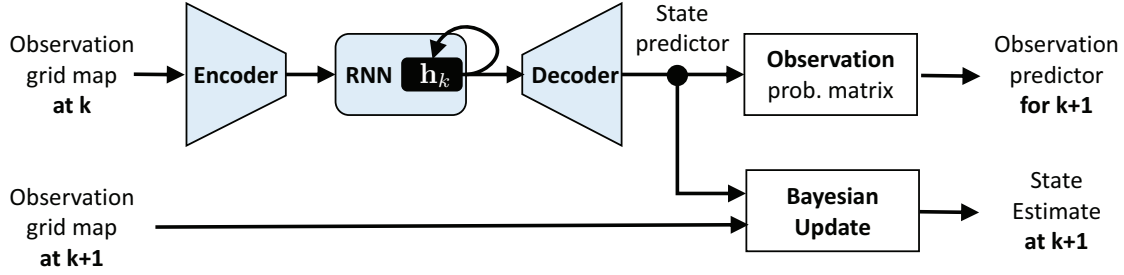


Fig. 2: Spatio-temporal state estimation network. The dynamic auto-encoder (colored light blue) is trained to predict the observation at $k + 1$ using the observation at k and the state of RNN denoted as h_k .

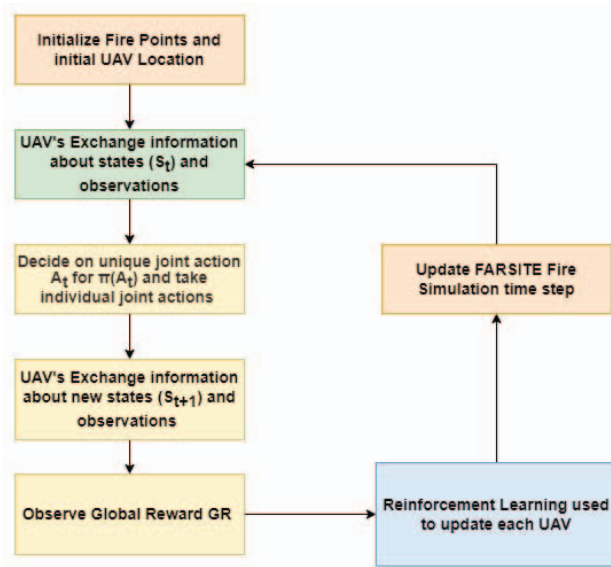


Fig. 3: Generalized methodology overview for different algorithms

fire points, UAVs and learning methods are initialized. Then the UAVs exchange information about the current state S_t and observations. The UAVs decide on a unique joint action A_t for policy π and take individual joint actions. After taking the individual joint actions, the UAVs exchange information about new states S_{t+1} . After exchanging information, a Global Reward GR is observed. After a global reward is observed, each method will perform an update using Reinforcement Learning for each UAV via Q-Learning, Q-Learning with Experience replay, DQN, and DQN with State Estimator. After the learning is updated, the FARSITE fire simulation time step is updated and this is repeated for 1000 time steps.

V. EXPERIMENTAL RESULTS

The simulation was setup on a $15 \times 15 \times 5$ discretized 3-D space with the FARSITE parameters mentioned in III. The simulation composed of $n = 3$ UAVs, with a total of 6 possible actions for each UAV: North, South, East, West, UP and Down. Each UAV in the team will receive a reward between 0 and 1 based on the number of Fire cells that the team of UAVs covers. While [14] considered overlaps to be negative and rewarded the team with a reward of 0, we consider overlaps

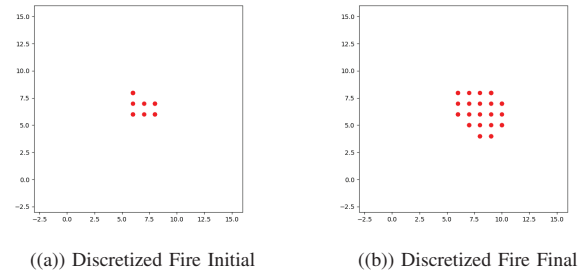
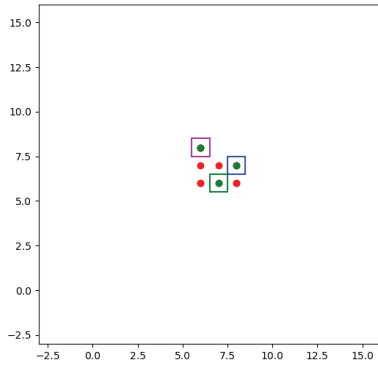


Fig. 4: 2-D view of a discretized fire simulation from the start to end of the simulation.

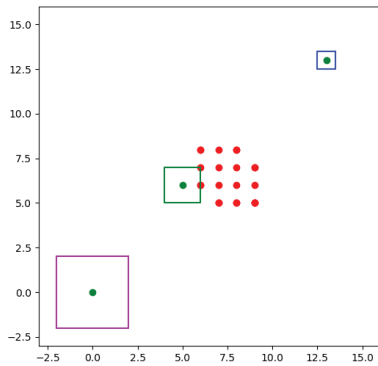
as redundant information but we do not punish the UAVs for having overlapping information. However, we do count overlapped fire cells as just one fire cell observed towards the total reward. There was a total of 1000 iterations for each episode, with a max reward of 1 per iterations, resulting in a max reward score of 1000 for an episode. The experimentation fire simulations were ran over 1000 episodes. Fig. 4 shows one of the simulation fire environments that was used to test and compare the algorithms. This figure shows the initial Fire environment state (Fig. 4 (a)) and the final Fire environment state (Fig. 4 (b)) in a discretized manner. The fire spreads from the initial to the final state as simulated using FARSITE, but the simulation is discretized in the 3-D space.

Fig. 5 shows the 2-D UAV coverage of a fire simulation as it learns the environment using the DQN and State Estimator. Fig. 5 (a) shows the initial deployment of the UAVs as they only cover 1 fire unit. As the fire spreads, the UAVs start the learning process. This can be seen on 5 (b) as only 1 UAV covers some fire units, while the remaining 2 UAVs are spread out and not covering any fire units. During the final deployment phases, the UAVs cover the entire fire as it spreads.

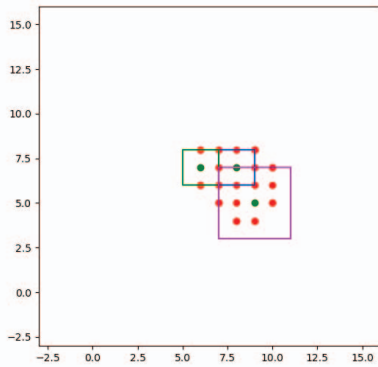
Using just Q-Learning the simulation results in a mean reward score of 150.1 and a median score of 137.5 as shown in Fig. 6. The UAVs do not reach the maximum reward of 1000, and the learning is not consistent, and there are constant changes in the reward of the team of UAVs. In Fig. 7, the Q-Learning and Experience replay approach is used. The simulation resulted in a mean reward score of 705.3 and a median of 782.0. Although there were increases in the mean and median reward



((a)) Initial Deployment of UAVs using the DQN and State Estimator Method.



((b)) UAV deployment learning the environment using the DQN and State Estimator Method.



((c)) Final deployment of the UAVs with total fire coverage using the DQN and State Estimator Method.

Fig. 5: 2-D UAV coverage of a discretized fire simulation as it spreads using the DQN and State Estimator Method.

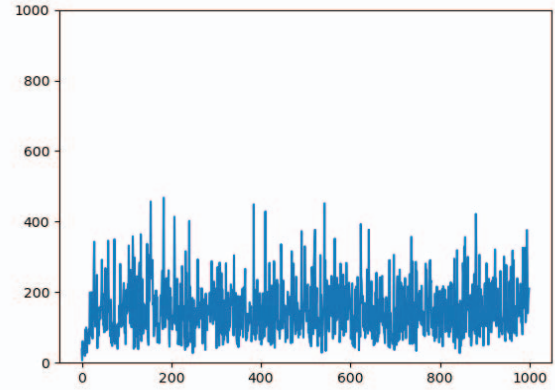


Fig. 6: Reward of the team of UAVs over 1000 episodes with Q-Learning. The X-axis represents the number of episodes and Y-axis represents the reward (max 1000).

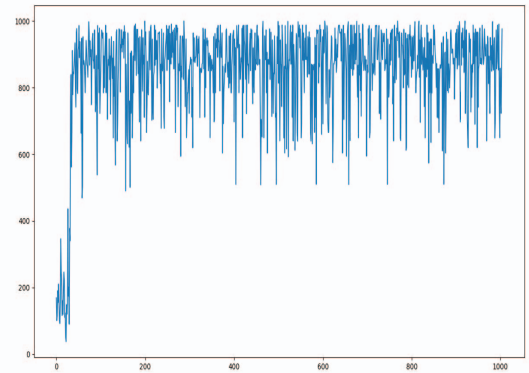


Fig. 7: Reward of the team of UAVs over 1000 episodes with Q-Learning and Experience Replay. The X-axis represents the number of episodes and Y-axis represents the reward (max 1000).

over 1000 episodes, there are constant changes in the reward of the team of UAV which shows a lack of convergence and constant learning with the approach. In Fig. 8, the DQN approach is used, and the simulation resulted in a mean reward score of 734.6 and a median of 783.0. We do not see much improvements from Q-Learning and Experience replay as the mean and median rewards over 1000 episodes are similar but the constant changes in the reward of the team of UAV still indicate a lack of convergence and constant learning with the approach. The approaches are still not consistent and concrete enough for the objective. However, in Fig. 9, the DQN and State Estimator approach is used, and the simulation resulted in a median reward score of 906.6 and a median reward score of 953. Additionally, the rewards show convergence as the team of UAVs consistently get similar reward scores after each episode. This indicates that while the UAVs are not able to cover the entire fire at all times, the UAVs are able to

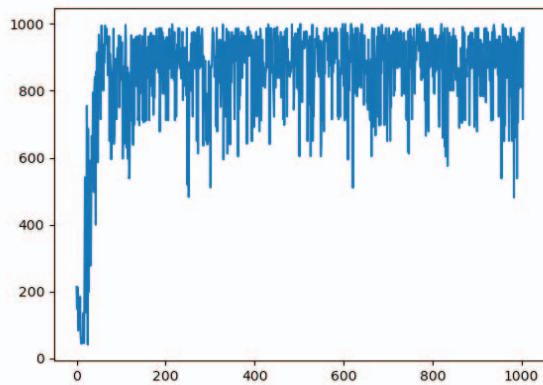


Fig. 8: Reward of the team of UAVs over 1000 episodes with DQN. The X-axis represents the number of episodes and Y-axis represents the reward (max 1000).

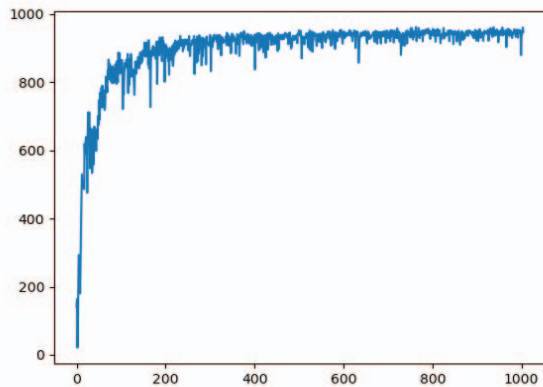


Fig. 9: Reward of the team of UAVs over 1000 episodes with DQN and State Estimator. The X-axis represents the number of episodes and Y-axis represents the reward (max 1000).

learn the environment and estimate the states and observations accurately and consistently. Other FARSITE fire simulations that we tested the DQN and state estimator method have shown similar results but with lower mean and median fire coverage. The model-based learning approaches have a lot more variance than the state estimator, and the results show that to be the case as the agents are still learning the environment and do not converge, even though the optimal reward is reached multiple times.

VI. CONCLUSION

In this paper, we implemented and evaluated 4 different methods and approaches for a cooperative and distributed approach for wildfire tracking and coverage with a team of UAVs. We implemented the Q-learning method from [14], took this Q-learning and added Experience Replay buffer to the learning method, a Deep-Q-Network and a Deep-Q-Network with a State Estimator. Preliminary results indicate that the Deep-Q-Network with a State Estimator solution shows significant

promise on this task and outperforms the Q-Learning, Q-Learning with an Experience Replay Buffer and Deep-Q-Network. The variances involved with just the model-based methods (Q-Learning, Q-Learning with an Experience Replay Buffer and Deep-Q-Network) are high and do not lead to convergence as the UAVs are still learning the environment. However, the proposed Deep-Q-Network with the State Estimator is promising as the team of UAVs are able to quickly converge, learn the environment and estimate the states and observations accurately and consistently. In the future, we plan to extend this work by varying the number of UAVs during simulation, considering more realistic physics to govern UAVs movement, and including more testing fire environments and simulations.

REFERENCES

- [1] "Wildfire statistics- congressional research service," <https://fas.org/sgp/crs/misc/IF10244.pdf>, last accessed: February 21, 2021.
- [2] M. Jafari, S. Sengupta, and H. M. La, "Adaptive flocking control of multiple unmanned ground vehicles by using a uav," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, Eds. Cham: Springer International Publishing, 2015, pp. 628–637.
- [3] A. C. Woods and H. M. La, "Dynamic target tracking and obstacle avoidance using a drone," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, Eds. Cham: Springer International Publishing, 2015, pp. 857–866.
- [4] Y. Alghamdi, A. Munir, and H. M. La, "Architecture, classification, and applications of contemporary unmanned aerial vehicles," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 9–20, 2021.
- [5] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, 2019.
- [6] H. X. Pham, H. M. La, D. Feil-Seifer, and L. V. Nguyen, "Autonomous uav navigation using reinforcement learning," 2018. [Online]. Available: <https://arxiv.org/abs/1801.05086>
- [7] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 4, pp. 665–676, 2019.
- [8] A. D. Dang, H. M. La, and J. Horn, "Distributed formation control for autonomous robots following desired shapes in noisy environment," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016, pp. 285–290.
- [9] H. M. La and W. Sheng, "Flocking control of multiple agents in noisy environments," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4964–4969.
- [10] M. T. Nguyen, H. M. La, and K. A. Teague, "Collaborative and compressed mobile sensing for data collection in distributed robotic networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1729–1740, 2018.
- [11] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, 2015.
- [12] H. M. La and W. Sheng, "Flocking control of a mobile sensor network to track and observe a moving target," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3129–3134.
- [13] —, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 766–778, 2013.
- [14] H. Pham, H. La, D. Feil-Seifer, and A. Nefian, "Cooperative and distributed reinforcement learning of drones for field coverage," *arXiv:1803.07250*, 09 2018.

- [15] A. C. Woods, H. M. Lay, and Q. P. Ha, "A novel extended potential field controller for use on aerial robots," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 286–291.
- [16] H. Pham, H. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6648–6653, 09 2017.
- [17] R. Dubey, S. J. Louis, and S. Sengupta, "Evolving dynamically reconfiguring uav-hosted mesh networks," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.
- [18] K. Shrestha, R. Dubey, A. Singandhupe, S. Louis, and H. La, "Multi objective uav network deployment for dynamic fire coverage," *2021 IEEE Congress on Evolutionary Computation (CEC)*, 06 2021.
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [20] R. Dubey, J. Ghantous, S. Louis, and S. Liu, "Evolutionary multi-objective optimization of real-time strategy micro," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 2018, pp. 1–8.
- [21] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [22] H.-J. Yoon, H. Kim, K. Shrestha, N. Hovakimyan, and P. Voulgaris, "Estimation and planning of exploration over grid map using a spatiotemporal model with incomplete state observations," *IEEE Conference on Control Technology and Applications, CCTA*, pp. 998–1003, 08 2021.
- [23] R. Konda, H. M. La, and J. Zhang, "Decentralized function approximated q-learning in multi-robot systems for predator avoidance," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6342–6349, 2020.
- [24] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, "Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach," *Automatica*, vol. 100, pp. 75–81, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109818305338>
- [25] D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using rrt," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 1429–1434.
- [26] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, "Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 680–691, 2017.
- [27] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military uav relay networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 804–807, 2016.
- [28] Y. Mualla, A. Najjar, A. Daoud, S. Galland, C. Nicolle, E. Shakshuki *et al.*, "Agent-based simulation of unmanned aerial vehicles in civilian applications: A systematic literature review and research directions," *Future Generation Computer Systems*, vol. 100, pp. 344–364, 2019.
- [29] F. Muñoz, E. S. Espinoza Quesada, H. M. La, S. Salazar, S. Commuri, and L. R. Garcia Carrillo, "Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 9, pp. 1566–1588, 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3687>
- [30] H. M. La and W. Sheng, "Dynamic target tracking and observing in a mobile sensor network," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996–1009, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889012000565>
- [31] C. Yuan, Z. Liu, and Y. Zhang, "Fire detection using infrared images for uav-based forest fire surveillance," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 567–572.
- [32] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 835–840.
- [33] L. Merino, F. Caballero, J. R. Martinez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *J. Field Robotics*, vol. 23, pp. 165–184, 03 2006.
- [34] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 639–643.
- [35] H. Cruz, M. Eckert, J. Meneses, and J. Martínez, "Efficient forest fire detection index for application in unmanned aerial systems (uass)," *Sensors (Basel, Switzerland)*, vol. 16, 2016.
- [36] D. Casbeer, R. W. Beard, T. McLain, S. Li, and R. Mehra, "Forest fire monitoring with multiple small uavs," *Proceedings of the 2005, American Control Conference, 2005.*, pp. 3530–3535 vol. 5, 2005.
- [37] C. Phan and H. H. T. Liu, "A cooperative uav/ugv platform for wildfire detection and fighting," in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, 2008, pp. 494–498.
- [38] I. Maza, F. Caballero, J. Capitan, J. R. Martinez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of Intelligent and Robotic Systems*, vol. 61, pp. 563–585, 01 2011.
- [39] C. Kyrkou and T. Theocharides, "Emergencynet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1687–1699, 2020.
- [40] M. Finney, "Farsite : Fire area simulator : model development and evaluation," 1998.
- [41] V. Krishnamurthy and G. G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp. 458–476, 2002.
- [42] H.-J. Yoon, D. Lee, and N. Hovakimyan, "Hidden Markov model estimation-based q-learning for partially observable Markov decision process," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2366–2371.
- [43] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012, vol. 26.
- [44] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. 12, 2010.
- [45] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014, pp. 1724–1734.
- [46] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2016.
- [47] S. Zhang and R. S. Sutton, "A deeper look at experience replay," in *Deep Reinforcement Learning Symposium, NIPS 2017*.