CSCI 5512 AI Project Proposal
Autonomous Wildfire Surveillance using Reinforcement Learning

Kerry Sun* and Peng Mun Siew†

1 Project Proposal

We propose to study how to navigate fixed-wing aircrafts autonomously to maximize forest fire coverage using deep reinforcement learning approaches. Specifically, given sensor information (aircraft states and partial wildfire image data), a deep neural network will be trained to maximize wildfire surveillance for single or pairs of aircraft. The research was originated from an recent AIAA conference paper [1]. We plan to mimic the study of the original work first, then improve models and compare different learning network. Specifically, a more detailed wildfire model (more tuning parameters) will be used as the environment for reinforcement learning. As for the agent, a simple dynamic aircraft model will be assessed in comparison to the simple kinematic aircraft model shown in the original paper. We plan to examine a Q-learning (shown in the original work) and a State-Action-Reward-State-Action (SARSA) network to observe the performance difference.

Wildfires consumed 10.1 millions acres of land in 2015 and cost an estimated $6 billion of damage from 1995 to 2004 [2]. Obtaining accurate information about the state of a wildfire during the course of its evolution is crucial in deciding where to use fire suppressants and where to remove fuel. Using Unmanned aerial vehicles (UAVs) can increase safety and bring economic benefits for wild fire monitoring. For example, there is no need of a real pilot to fly in dangerous condition if the UAV can be used in real time. Furthermore, multiple UAVs can operate together without human control, reducing costs and increase efficiency.

We will show some preliminary work of the environment and agent modeling as well as some proposed reinforcement learning methods in the following sections.

2 Wildfire Model

A wildfire model is needed to create a simulation environment to train and evaluate a controller. We use the wildfire model in [3]. The model was derived from the conservation of energy, balance of fuel supply and the fuel reaction rate:

$$\frac{dT}{dt} = \nabla \cdot (k\nabla T) - \overrightarrow{v} \cdot \nabla T + A\big(Se^{-B/(T-T_a)} - C(T - Ta)\big) \tag{1}$$

$$\frac{dS}{dt} = -C_S S e^{-B/(T-T_a)}, \quad T > T_a \tag{2}$$

with the initial values

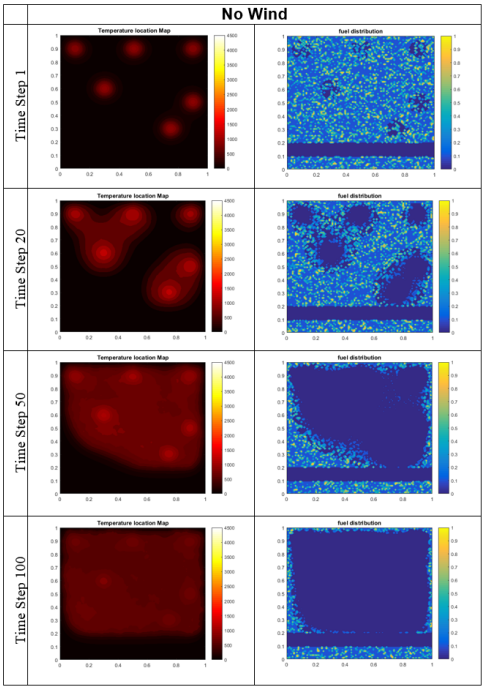
$$S(t_{init}) = 1 \quad and \quad T(t_{init}) = T_{init} \tag{3}$$

where $T(K)$ is the temperature of the fire layer. $S \in [0, 1]$ is the fuel supply mass fraction (the relative amount of fuel remaining), $k(m^2 s^{-1})$ is the thermal diffusivity, $A(Ks^{-1})$ is the temperature rise per second at the maximum burning rate with full initial fuel load and no cooling present, $B(K)$ is the proportionality coefficient in the modified Arrhenius law, $C(K^{-1})$ is the scaled coefficient of the

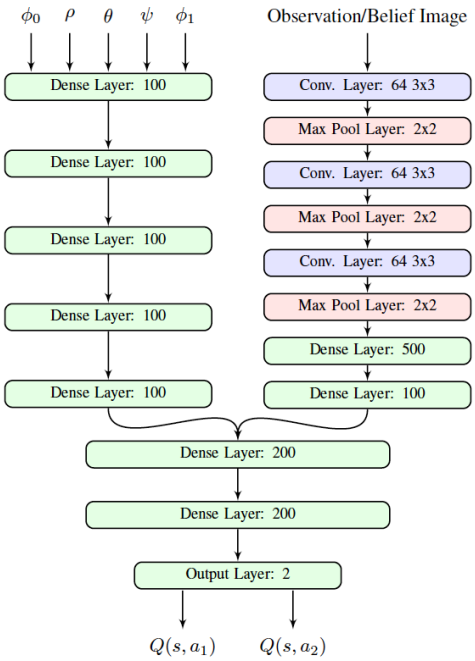
*Department of Aerospace Engineering and Mechanics, sunx0486@umn.edu
†Department of Aerospace Engineering and Mechanics, siewx007@umn.edu

heat transfer to the environment, $C_s(s^{-1})$ is the fuel relative disappearance rate, $T_a(K)$ is the ambient temperature, and finally $\overrightarrow{v}(ms^{-1})$ is wind speed.

For simulation, we construct a spatial discretization using Laplacian matrix in Matlab. The backward Euler and Godunov splitting were used for the time stepping. Also, the wildfire model is with randomized fuel distribution to make it more realistic. Fuel map and Fire Temperature maps are recorded at each time step. Those maps will be used as observations for the reinforcement learning networks. Fig. 1(a) is an example of wildfire propagation over 100 iterations for no wind conditions from our simulation.



(a) Wildfire propagation over time



(b) Network Architecture [1]

Figure 1: Wildfire Simulation and Network Architecture Example.

3 Agent Modeling

Given an agent in state $s \in S$ taking action $a \in A$, the agent will transition to new state $s' \in S$ and receive reward $r \in R$ based on a Markov decision process. In this application, the agent is a UAV being controlled, the actions will dictate the aircraft's trajectory. Since the full state of the wildfire is unknown, decisions must be made using partial observations, and this is known as partially observable Markov decision process (POMDP). We will either have one or 2 aircrafts for this study.

3.1 Aircraft State

We will use similar aircrafts states from [1] for our decision making. Those states are:

1. ϕ_0 bank angle of aircraft 1
2. ρ range to aircraft 2
3. θ bearing angle to aircraft 2 relative to current heading direction of aircraft 1
4. ϕ heading angle of aircraft 2 relative to current heading of aircraft 1
5. ϕ_1 bank angle of aircraft 2

We may also include other states since a different dynamic model maybe implemented. We will use similar actions and rewards from [1] for our agents.

3.2 Dynamics

Since the dynamic model of the aircraft in [1] is too simple, we will use a full aircraft dynamic model for the UAVs. Those differential equations can be found in Chapter 3 of [4].

By combining the wildfire environment and the dynamics of the aircrafts, we can create an platform to train the agent to maximize its reward to monitor wildfire. Fig. 2 shows an example of an aicraft's wildfire observation (taken from [1]).

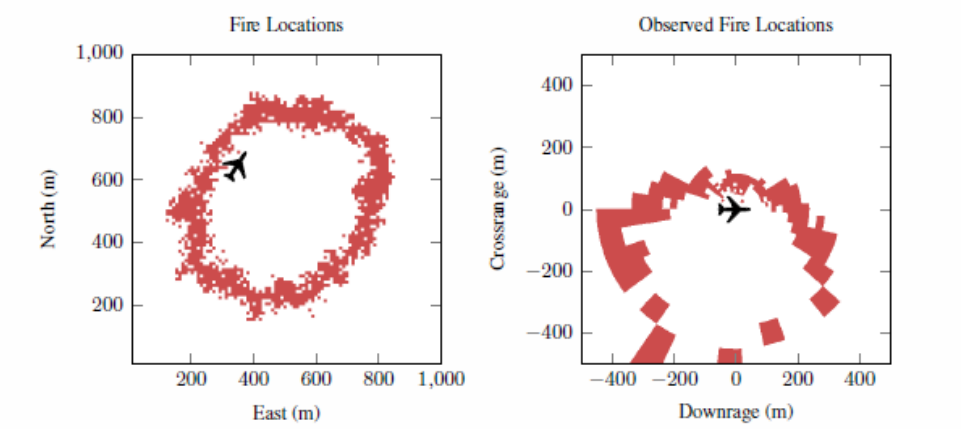


Figure 2: Example of an aircraft's wildfire observation [1]

4 Reinforcement Learning Network

The reinforcement learning network will be constructed using TensorFlow [5], which is an open-source software library for dataflow programming across a range of tasks.

The deep Q-network architecture proposed by Julian and Mykel as shown in Fig. 1(b) will be used as a benchmark architecture to compare the performance of our proposed architecture [1]. In the paper, a deep Q learning network is used to learn the optimum decentralized controller to maximize forest fire coverage for multiple fixed-wing aircraft.

In our proposed architecture, we plan to examine the performance of the SARSA [6] network in this application compared to the Q Learning network used in the paper. The main difference between the two approaches is how the Q-value of each state-action pair is estimated. The Q-value represents the expected rewards for taking a particular action from a particular state. The Q Learning network uses the Bellman equation to estimate the Q-value, where it is assumed that the optimal action will be taken in the next state, whereas in the SARSA implementation, the agent is assumed to follow the current policy to dictate the next action when estimating the Q-value of the current state-action pair. The Q-value update equations are shown in Eq. (4) and Eq. (5) below for the Q Learning network and the SARSA network respectively.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)] \tag{4}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{5}$$

Furthermore, we also plan to look into the effect of varying network depth and network architecture on the performance of the network, such as introducing drop out layers and batch normalization to prevent data overfitting. Besides that, in the recent years, residual network (ResNet) has been garnering attention of the deep learning community due to its ease of training and the increase in accuracy with increasing depth. ResNet has shown to outperform other network in ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation [7].

References

[1] K. D. Julian and M. J. Kochenderfer, "Autonomous distributed wildfire surveillance using deep reinforcement learning," in *AIAA SciTech Forum*. American Institute of Aeronautics and

Astronautics, Jan. 2018. [Online]. Available: https://doi.org/10.2514/6.2018-1589

[2] "Fereral firefighting costs (supression only)," National Interagency Fire Center, 2017.

[3] J. Mande, L. S. .Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek, "A wildland fire model with data assimilation," *Mathematics and Computers in Simulation*, vol. 79, no. 3, pp. 584–606, Dec. 2008.

[4] V. Klein and E. A. Morelli, *Aircraft System Identification Theory and Practice*, J. A. Schetz, Ed. AIAA Education Series, Aug. 2006.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Bradford, Ed. Cambridge, Massachusetts: The MIT Press, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.