FireBrake: Optimal Firebreak Placements for Active Fires using Deep Reinforcement Learning

Aadi Kenchammana
Saint Francis High School
Mountain View, CA 94024
aadikenchammana@sfhs.com

Abstract

As wildfires grow in intensity and frequency, generating effective placements for firebreaks - obstacles to wildfire spread - is an urgent need for wildfire management. In this paper, I propose FireBrake, a system with two components: (1) a wildfire simulator that incorporates realistic wind and moisture data and (2) a Deep Reinforcement Learning (DRL)-based firebreak placement solution. Unlike previous work, FireBrake is evaluated against real fires using two metrics: simulation accuracy and burnt area reduction. Experiments using real fire data show that FireBrake outperforms existing solutions both in terms of simulation accuracy (1.8 -5.5% error reduction in estimated burnt area compared to previous simulators) and mitigation efficacy (4-8% reduction in burnt area compared to real firefighters). The codebase is available at https://github.com/Aadikenchammana/FireBrake.

1 Introduction

Wildfires are one of the most destructive natural phenomena worldwide and have been increasing in length and severity [1]. In 2020 alone, they caused over 16.5 billion dollars in damage to insured property and over 93 billion dollars in damage from reduced economic productivity in the United States alone [2]. Firebreaks, barriers that impede the spread of wildfires, are an effective mitigation strategy, but they are time-consuming and expensive to build. Additionally, it is difficult for firefighters to decide where to place firebreaks during an unfolding crisis. Developing an effective computational solution for firebreak placement is an urgent need for wildfire management.

Previous work on effective firebreak placements has utilized Genetic Algorithms (GAs) [3], which are computationally expensive and challenging to generalize. Furthermore, previous work on fire prevention for simulated fires showed that GAs underperformed Deep Reinforcement Learning (DRL)-based approaches [4]. DRL has previously been used to develop solutions for complex, high-dimensional scenarios such as Atari video games[5], chip design[6], and language models [7]. This makes it a more promising candidate for generating firebreak placement. A recent paper [8] introduced a DRL-based approach to place firebreaks in a developing wildfire. The paper introduced SimFire, a wildfire simulator, and SimHarness, a DRL-based training framework for firebreak placement. SimFire simulated wildfire spread based on fuel, wind, and topography data using the Rothermel Fire Model [9]. SimHarness trained a Deep Q Network (DQN) agent to place firebreaks in a developing fire. While the approaches in this paper are a step in the right direction, they have several limitations. SimFire didn't accurately model key predictors of fire spread - wind and live fuel moisture content (LFMC), the percentage of biomass made up of water. It utilized the Perlin Noise Algorithm to pseudo-randomly generate wind data, a simplistic approach that cannot accurately model the localized wind patterns that influence fire behavior. Similarly, it assumed a static LFMC, even though LFMC varies dynamically between forest regions [10]. SimHarness used a simplistic state space that only described the current fire spread, not the risk of future spread, and only implemented

Preprint. Under review.

DQN, which can struggle to converge on an optimal solution [11], instead of other DRL algorithms such as Proximal Policy Optimization (PPO). Additionally, SimHarness was only evaluated against simulated fires.

In this paper, I present FireBrake which implements a more advanced wildfire simulator and a DRL-based algorithm for firebreak placement. The FireBrake simulator is an improvement over previous work and incorporates a realistic wind simulator and LFMC data. The FireBrake placement algorithm utilizes a more realistic state and improved reward functions with multiple reinforcement algorithms, PPO and DQN, to control an active fire. Unlike previous work, FireBrake's efficacy was evaluated through comparisons to real fire data obtained from the BurnMD dataset [10].

2 FireBrake - Methods and Evaluation

2.1 FireBrake Wildfire Simulator

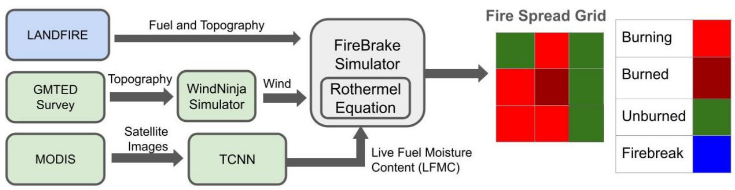


Figure 1: FireBrake Simulator; the green boxes show my additions to the SimFire system, shown in blue.

The FireBrake simulator (Fig 1) simulates the spread of a wildfire in a discrete environment by using the Rothermel Fire equation [9], a well-established model for calculating the rate of wildfire spread. It incorporates basic fuel and topography data from USFS's LANDFIRE [12] database; wind data, simulated with Firelab's WindNinja Simulator[13]; and LFMC data, estimated from MODIS satellite data [14]. The wildfire area is represented as an H x W grid where all squares are initialized to unburned (0), except the ignition point. The Rothermel Fire equation is used to calculate the rate of fire spread from each ignited square to its adjacent squares; when the value at a square exceeds a certain threshold, it is set to burning (1), After a set number of timesteps, a burning square transitions to burned (2). If a square has a firebreak, it cab never transition to burning or burned.

The WindNinja [13] simulator models localized wind patterns using Computational Fluid Dynamics and by taking topography into account. Localized topographical data is acquired from the Global Multi-Resolution Terrain Elevation Data (GMTED) survey with a resolution of 250m by 250m [15]. The LFMC data is estimated using an existing Temporal Convolutional Neural Network (TCNN) model [16] based on remote sensing data products from NASA's MODIS satellite [14], NASA's SRTM mission [17], and OSU's PRISM AN81d dataset [18]. Since the published LFMC model was trained using a Köppen-Geiger climate zone classification feature, which is not available for most fire areas, I retrained the model excluding this feature using the Globe-LFMC dataset [19], which catalogs LFMC at several sites, as the target feature. The quality of the new LFMC model, as measured by correlation, was similar to that of the original model.

2.2 FireBrake Placement Solution

The goal of the FireBrake placement solution is to optimally place firebreaks in a developing wildfire such that burnt area is minimized. This is accomplished by training a DRL agent (model) using Stable_Baselines3 [20], a Python DRL library. A DRL problem is formulated mathematically as a Markov Decision Process [21], where, at each timestep t, the environment is in a state s and the agent takes action a to maximize the reward r. The environment transitions to a new state s' according to environmental dynamics. The agent (model) learns the policy π that maps a state to the optimal action to maximize r. For the FireBrake placement algorithm, the different components were defined as follows (Fig 2A):

- Each state s is represented as a H x W x C grid representing the fire area with H = 27 pixels, W = 27 pixels, and C=2 channels. The first channel describes burn status (B_s) , which

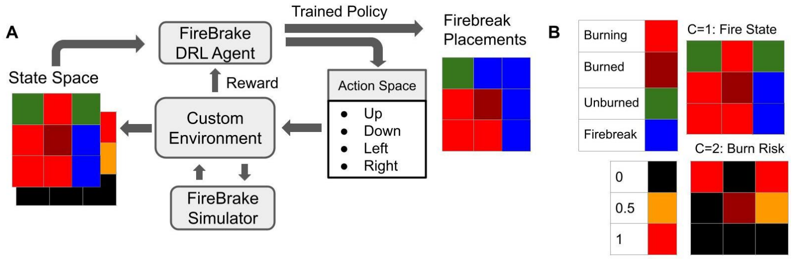


Figure 2: FireBrake Placement Solution: (A) DRL Agent Training Framework (B) Example State Space.

represents each square as burned, burning, unburned, or having a firebreak. The second channel describes burn risk (B_r) which represents how soon a square might ignite. $B_r = \frac{1}{t_i}$ where t_i is the number of timesteps to ignition for that square and $0 \leq B_r \leq 1$.

- Each action $a \in \{\text{up, down, left, right}\}$. A mitigation is placed after each action.
- Several reward functions are available in our system: R_{Dist} measures the proportion of area burned and the agent's distance from the fire; R_{vel} incorporates the rate of fire spread into R_{Dist}; and R_{bench} is the reward function implemented by SimHarness.
  - $R_{Dist} = -10\frac{Damaged_{sim_t}}{Area_{total}} - 0.3Dist_{0.5r}$,
  - $R_{vel} = -10\frac{Damaged_{sim_t}}{Area_{total}} - 0.3Dist_{0.5r} - 10\frac{Damaged_{sim_t} - Damaged_{sim_{t-1}}}{Damaged_{sim_{t-1}}}$,
  - $R_{bench} = \frac{(Damaged_{sim_t} - Damaged_{sim_{t-1}}) - (Damaged_{bench_t} - Damaged_{bench_{t-1}})}{Damaged_{bench_{final}}}$

  where $Dist_{0.5r}$ is the agent's distance to the closest square with $B_r = 0.5$, $Damaged_{sim_t}$($Damaged_{bench_t}$) is total burned+burning area with (without) FireBrake mitigations at timestep *t*.
- The policy π is a Multi-layer perceptron (MLP) network, containing 128 units in two fully-connected layers, that is trained to select the best candidate action given a state.

In our framework, the MLP is trained using either DQN or PPO. First, a general DRL agent is trained over 30,000 episodes by randomly changing the fire ignition point every 2,500 episodes. To place firebreaks in a given fire, the general agent is then fine-tuned over 10,000 episodes by starting with the original ignition point and keeping it unchanged throughout the fine-tuning.

2.3 Evaluation

FireBrake is evaluated against real fires and firebreaks from the BurnMD dataset [22]. To test the accuracy of the FireBrake wildfire simulator, the fire's spread is simulated up to the time when the first firebreaks were placed and the actual fire spread (polygon area cataloged in BurnMD) is compared to the simulated area. To assess the FireBrake placement solution, the burnt area obtained after firefighter mitigations are placed (final polygon area of the fire) is compared to the burnt area with FireBrake mitigations. The firebreaks for each real fire are placed using a FireBrake agent is fine-tuned using the initial position of the real fire as obtained from BurnMD.

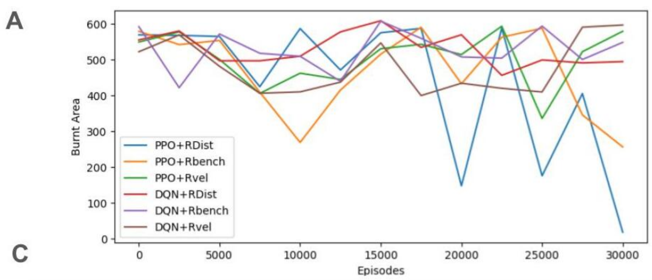
3 Results

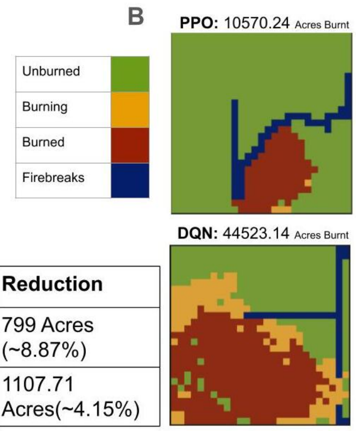
FireBrake Simulator: To test FireBrake simulator accuracy, the simulation results were compared to the real spread prior to mitigation placements, and the results of SimFire. To ensure a fair comparison, both simulators used the same number of timesteps, which approximated the time before the first mitigation was placed. As shown in Figure 3, Firebrake has a 1.8-5.5% reduction in error compared to SimFire for two real fires.

FireBrake Placement Solution: First, I trained PPO and DQN agents using all three reward functions on an AWS M7a.4xLarge instance for 6 - 10 hours. The PPO agent trained using R_{Dist} outperformed

	SimFire: Predicted Burn Area	FireBrake: Predicted Burn Area	BurnMD: Actual Burn Area
2020 Medicine Wildfire (AZ) @T = 96 hr	Burn Area: 9742 Acres Error: 10.44%	Burn Area: 9243 acres Error: 4.79%	Burn Area: 8820 acres
2020 Slink Wildfire (CA) @T = 24 hr	Burn Area: 4998 acres Error: 5.17%	Burn area: 4909 Acres Error 3.33%	Burn Area 4752 Acres

Figure 3: FireBrake Simulator Results; Comparison of burnt areas simulated by SimFire and FireBrake to the real burnt area from BurnMD.





C

	FireBrake Burn Area	BurnMD Burn Area	Reduction
2020 Medicine Wildfire (AZ)	8202.Acres	9001 Acres	799 Acres (~8.87%)
2020 Slink Wildfire (CA)	25592 Acres	26708 Acres	1107.71 Acres(~4.15%)

Figure 4: FireBrake Placement Results: (A) Total burnt area for DRL algorithm and reward function combinations sampled every 2500 episodes (B) Comparison of PPO and DQN agent trained with R_{dist} towards the end of training (C) Comparison of burnt area of real fires with FireBrake mitigations and real mitigations.

all other combinations of reward functions and agent types in terms of burnt area reduction at the end of training (Fig 4A). Notably, this configuration outperformed the DQN agent trained using R_{Bench} which is what SimHarness proposed (despite multiple attempts, I was unable to get the SimHarness code working). Figure 4B shows that the best PPO agent successfully built a perimeter around the fire and blocked its advance while the best DQN agent, also trained with R_{Dist}, failed to build effective firebreaks and contain the fire's spread. This improvement is likely attributable to PPO being an on-policy model, meaning it optimizes the policy as a whole. DQN is off-policy meaning it optimizes the Q-Function which is a substitute for the policy. This makes PPO more effective in complex environments. The best-performing agent, PPO with R_{dist}, was then evaluated against firefighter response in both the 2020 Slink and Medicine wildfires. Figure 4C shows that the FireBrake agent was able to cause a meaningful reduction, 4% to 8%, in burnt area. Additional results are available at https://github.com/Aadikenchammana/FireBrake/tree/main/supplemental_data.

4 Conclusion

In this paper, I propose FireBrake, which consists of a wildfire simulator and a DRL-based firebreak placement solution. By implementing the WindNinja simulator and more accurate LFMC data, the FireBrake fire simulator achieved a 1.8%-5.5% reduction in error rate compared to SimFire. The FireBrake placement solution can train multiple different types of agents with different reward functions. The best performing configuration was a PPO agent trained with R_{Dist} which outperformed SimHarness and was able to reduce the total burned area of wildfires by 4%-8% when compared to the actions of real firefighters.

In the future, I plan to continue improving both the FireBrake simulator and the FireBrake placement solution. I will improve the the simulator by incorporating new ML-based fire modelling techniques. The FireBrake placement solution can be extended to include multiple agents, which would more closely emulate the response to most major fires, and account for variable travel and firebreak construction times in heterogeneous terrain and weather conditions.

5 References

1. Flannigan, Mike, et al. 2013. "Global wildland fire season severity in the 21st century". Forest Ecology and Management vol 294. https://www.sciencedirect.com/science/article/abs/pii/S0378112712006196.
2. Bayham, Jude, et al. 2022. “The Economics of Wildfire in the United States”. The Annual Review of Resource Economics. pp. 379 - 401. .https://www.fs.usda.gov/rm/pubs_journals/2022/rmrs_2022_bayham_j001.pdf
3. Homchaudhuri Baisravan, Kumar Manish, Cohen Kelly. 2010. “*Wildfire Fighting in Uncertain and Heterogeneous Environment”*. Proceedings of the American Control Conference. https://ieeexplore.ieee.org/document/5531049
4. Altamimi Abdulelah, et al. 2022. “*Large-Scale Wildfire Mitigation through Deep Reinforcement Learning”*. Front. For. Glob. Change (vol 5). https://www.frontiersin.org/articles/10.3389/ffgc.2022.734330/full
5. Mnih, Vlodymyr, et al. 2013. "Playing Atari with Deep Reinforcement Learning". NeurIPS Deep Learning Workshop 2013. https://arxiv.org/abs/1312.5602
6. Mirhoseini, Azalia, et al. 2021. "A Graph Placement Method for Fast Chip Design". Nature. https://www.nature.com/articles/s41586-021-03544-w
7. Ouyang, Long, et al. 2022. "Training language models to follow instructions with human feedback". NeurIPS. https://arxiv.org/abs/2203.02155.
8. Tapley Alexander, et al. 2023. “*Reinforcement Learning for Wildfire Mitigation in Simulated Disaster Environments”*. NeurIPS. https://arxiv.org/abs/2311.15925
9. Andrews, Patricia L. 2018. The Rothermel surface fire spread model and associated developments: A comprehensive explanation. Gen. Tech. Rep. RMRS-GTR-371. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 121 p.
10. Camprubi, Angel C., et al. 2022. "Live Fuel Moisture Content Mapping in the Mediterranean Basin Using Random Forests and Combining MODIS Spectral and Thermal Data". Remote Sensing. https://www.mdpi.com/2072-4292/14/13/3162
11. Van Hasselt, Hado, et al. 2016. "Deep Reinforcement Learning with Double Q-Learning". *Proceedings of the AAAI Conference on Artificial Intelligence*. https://arxiv.org/abs/2306.01451.
12. Rollins Matthew G., et al. 2009. *“LANDFIRE: a nationally consistent vegetation, wildland fire, and fuel assessment”*. International Journal of Wildland Fire. https://landfire.gov/documents/Rollins2009landfire.pdf
13. Wagenbrenner, Natalie S., et al. 2015. "Downscaling surface wind predictions from numerical weather prediction models in complex terrain with WindNinja". Atmospheric Chemistry and Physics. https://weather.firelab.org/windninja/publications/downscale.pdf
14. https://modis.gsfc.nasa.gov/data/. "MODIS". Maccherone, Brandon.
15. https://www.usgs.gov/coastal-changes-and-impacts/gmted2010. "GMTED2010". Danielson, Jeffery J.
16. Miller, Lynn, et al. 2022. "Multi-Modal Temporal Cnns for Live Fuel Moisture Content Estimation". Environmental Modelling & Software. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4118154.
17. https://www.earthdata.nasa.gov/sensors/srtm. "Shuttle Radar Topography Mission (SRTM)".
18. https://prism.oregonstate.edu/. "PRISM Climate Data".
19. Yebra, Marta, et al. 2024. "Globe-LFMC 2.0, an enhanced and updated dataset for live fuel moisture content research". Sci Data 11. https://www.nature.com/articles/s41597-024-03159-6
20. https://stable-baselines3.readthedocs.io/en/master/. "Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations".
21. Otterlo, Martijn & Wiering, Marco. 2012." Reinforcement Learning and Markov Decision Processes. Reinforcement Learning". Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. https://doi.org/10.1007/978-3-642-27645-3_1

22. Dotter, Marissa. 2023. *"A Fire Projection and Mitigation Modeling Dataset*". ICLR 2023. https://www.climatechange.ai/papers/iclr2023/19