Simulating Responsive Action to Wildfires

Nathan Pannell
University of Victoria
npannell@uvic.ca

Triston Grayston
University of Victoria
tristongrayston@uvic.ca

Finn Archinuk
University of Alberta
farchinu@ualberta.ca

Mohammad Chegini
University of Victoria
mchegini@uvic.ca

Ashley Philip
University of Victoria
ashleyphilip07@uvic.ca

***Abstract*—This work introduces the novel approach of including active suppression by incorporating artificially intelligent agents to a wildfire propagation environment. By employing AI-driven agents within a simulation environment, this research assesses the efficacy of different containment tactics. The simulation utilizes multiple data layers aiming to replicate realistic fire spread scenarios by taking into account wind, fuel moisture, and fire intensity. Initial findings demonstrate a significant improvement in fire containment through strategic AI interventions. Our work underscores the need for further advancements in this area, including the integration of real-time environmental data and the further exploration of agent-based suppression strategies.**

I. Introduction

The 2023 wildfire season in British Columbia marked an unprecedented challenge, highlighting the critical need for revamping traditional wildfire combat strategies. This paper explores the integration of advanced technologies into wildfire management, particularly the utilization of Artificial Intelligence (AI) as proposed by the Canadian Wildfire Strategic Plan (2022 – 2027).

Our research delves into the development of sophisticated simulation models that incorporate these environmental factors, employing AI agents that simulate traditional water-based firefighting techniques. These simulations aim to evaluate the potential of various containment tactics and develop a strategic, AI-driven approach for moisture application, informed by predictive analyses of at-risk areas.

Underpinning the operational advancements, this paper details the application of the Proximal Policy Optimization algorithm [Schulman et al., 2017] and an elaborate reward structure within the AI agents' decision-making processes. Such innovations in the computational modeling of wildfire management represent a shift towards a more analytical and preemptive strategy, promising significant improvements in fire containment efforts. The integration of AI into the execution of established tactics holds the potential to enhance resource allocation efficiency, thereby contributing to more effective management of future wildfire events.

A. Motivation

In recent years, it has become apparent that wildfires have become increased in frequency and destruction across British Columbia, with the 2023 fire season being the most destructive in B.C's recorded history. More than 2.84 million hectares of land was burned, which resulted in the evacuation of approximately 48,000 people and for the province of B.C to declare a state of provincial emergency for a total of 28 days [BC Government Summary, 2023]. These fires are destructive to land and infrastructure, and the resulting smoke has major impacts to human health. Additionally, the decrease in air quality caused by fires leads to increased asthma and COPD exacerbation's [Reid et al., 2016].

The goal of our project is to join modern AI techniques to established wildland firefighting practices resulting in better tools for fire spread forecasting and resource management.

B. Problem Definition

There has been a boom of AI applications into many industries, including wildfire prevention and suppression. The Canadian Wildfire Strategic Plan (2022-2027) aims to implement AI/ML into wildland firefighting [BC Government Summary, 2023]. This plan focuses on many aspects of wildfire including data analysis, response methods, response predictions, and improvements to mental and physical health.

We note that the technology being introduced into the field focuses primarily on large scale forest fires. Because of this, 2 key factors are often overlooked, which includes how drastically the actions of wildland firefighters can effect smaller fires, and the effects of *spotting* behaviour [Fire manual, 2016], when embers jump moist portions of land creating new fires outside of the prior perimeter.

Our project contributes the following:

1) Uses various heuristics to highlight the dramatic effects initial attack crews have on wildfires.
2) Focuses on smaller fires no larger than 4.5km x 4.5km.
3) Encapsulates the importance of spotting behaviour.

II. Related Work

Early wildland fire spreading computer simulations utilized grid-based representation of different physical parameters important for predicting the propagation of fire spread [Kourtz et al., 1977]. This work, owing to the computational burden and the time limitation, relies on a simplified model of fire spread, where burning cells can ignite their neighbours given sufficient conditions have been met.With these computational limitations, these early models focused on fuel type, wind, and fuel moisture.

A modern approach that follows this style of representation is Cell2Fire [Pais et al., 2019]. One of the major contributions of Cell2Fire's approach is that a cell can only be updated by its eight immediate neighbours. By limiting more distant interactions computational overhead was reduced, which was offset by rapid update time steps on the order of minutes. Propagation to these neighbours is affected by the wind direction, where ellipses are used to determine the updating of neighbours. A major limitation of their work is that fire is unable to breach natural barriers, including but not limited to bodies of water, i.e. rivers or ponds, fields, and gravel pits. This is a deterministic model, with variation available through different meteorological conditions and the initial location of the ignition.

Prometheus has been Canada's fire spread simulator for over 20 years and is now being phased out [Tymstra et al., 2010]. It uses a polygon-based fire front where vertices of the polygon are updated following an elliptical model. In the literature, this is referred to as wave propagation. The fire front is determined to be the hull connecting the new front and edges of the polygon are rediscretized for future steps. It is worth mentioning that unlike Cell2Fire, Prometheus is able to display *breaching* behaviour, where the flames of a fire can jump across the aforementioned barriers.

Prometheus is unable to model different intensities of fire, with the fire front indicating either on-fire or not. As we are interested in agent-based suppression of fires, we require more detailed simulations. Prometheus is primarily focused on the burn scar - the area burned by an uncontrolled fire.

Most simulation softwares consider fire spread as a two-dimensional problem. Farguell et al. [Farguell et al., 2019] look at the effects on wind as a function of topology of the region and heat of the ongoing fire, both of which affect local wind direction and magnitude. They simplify this interaction using a more coarse atmospheric grid.

Upon reviewing the literature, we discovered that there has been no attempt to model active fire suppression as part of the simulation of fire spread.

III. Methodology

The project can be fundamentally partitioned into two primary components: Simulation and Agents.

A. Simulation

Simulation was written in Python heavily utilizing the NumPy and SciPy packages. Inspired by the different information layers in Cell2Fire, our simulation includes a base 4.5km×4.5km grid divided into 90×90 cells. Furthermore, as shown in Figure 1, our simulation utilizes multiple channels for different attributes, including fire intensity, available fuel, and moisture level. Given the small-scale focus, a single wind vector and humidity scalar are used to represent the local values for the entire map. Also, we differentiated the fire

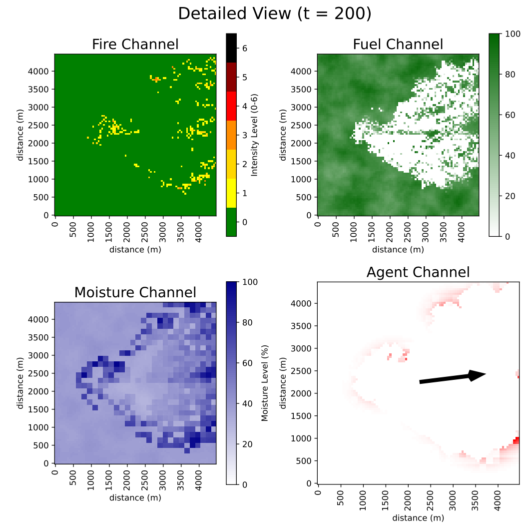

Fig. 1: Split Channel View of Simulation. Fire intensity is discretized to 6 ranks. Fuel shows the coordinates of remaining vegetation. Moisture slows fire propagation. Wind direction and magnitude is shown in the agent channel. This agent probabilistically selects where to suppress a fire based on wind and active fire locations.

intensity into 6 ranks (informed by [Fire manual, 2016]), which impact the fuel consumption rate, fire spread likelihood, and intensity of new fires. Higher fire ranks represent more intense fires and thus increase fuel consumption as well as the chance of contagion.

Fire rank was limited by fuel availability. As the fuel in each cell approaches 0, the intensity of the fire is clipped to a successively lower ceiling. At 10%, 20%, 30%, and 40%, the fire rank is clipped to 1, 2, 3, and 4 respectively. The reasoning is that a 50m×50m area with greater than 40% of the area containing fuel should be able to sustain a fire of any size, while a lower amount of fuel would result in smaller ranks. Intuitively, this represents the exhaustion of fuel that would support large fires and allow the continuation of smaller embers.

Our simulator is discretized in time with resolution set to one day. Other simulators operate at a finer time resolution, but our approach mirrors the realistic activity of a firefighting team initial attack (IA) crew, making an impact on an area of 150m×150m per day .

B. Kernel-based Updating

We utilize convolutional kernels both for updating the simulation and prioritizing regions of the environment for the heuristic agent. The application of a kernel is done through a convolution operation that is easily parallelizable.

Generating the kernel starts with a 2D Gaussian distribution to prioritize more significant cells by spatial distance. The wind magnitude is calculated for the time step, which we use to initialize the covariance matrix. We subsequently rotate the distribution by the wind direction. Rotating the distribution involves rotating the covariance matrix, and correcting that rotation using the transpose of that matrix:

$$\Sigma' = R\Sigma R^T$$

where Σ is the original covariance matrix, and R and R^T produces the corrected rotation. The distribution is then rasterized to the appropriate spatial resolution. The flame kernel is 7×7 (which translated into 350m×350m) and when convolved with the fire intensity layer results in an ignition map. This ignition map is a probabilistic representation of which cells could start burning in the following time step. Because we are using a probabilistic non-adjacent update, we naturally introduce the possibility of embers jumping from active fires to disconnected regions.

The heat kernel is rasterized to 30×30 (1.5km×1.5km) and also operates on the fire intensity layer. This results in an evaporation map, decreasing the moisture layer, which makes the ignition of this cell more likely in later time steps. The heat kernel is reused as input for the heuristic agent as it indicates where the fire is likely to spread in subsequent time steps.

Algorithm 1 Fire Spread Update Calculation

ignitions = zero matrix$(mapSize \times mapSize)$
for $i = 1$ to mapSize **do**
  for $j = 1$ to mapSize **do**
    noise $\sim$ Uniform$(0, 1)$
    dryness $= 1 -$ moisture$_{i,j}$
    ignitionProb = ignitionMap$_{i,j}$ $\times$ dryness
    if noise $<$ ignitionProb **then**
      ignitions$_{i,j} = 1$
    else
      ignitions$_{i,j} = 0$
    end if
  end for
end for

C. Agents

In real life wildfire control environments, there are different types of crews used based on the size of the fire, and how the fire can best be approached [Fire manual, 2016]. For less intensely burning fires (rank 1-3) which require helicopter access [BC Government Crews, 2023], initial attack crews which consist of 3-5 members are sent to take action. Our system models this behavior, with each of the agents representing the combined work of a initial attack team.

The intention of the agent is to understand what the dynamics of what is happening on the board and provide water to a cell in order to make that cell either less susceptible to igniting, or decreasing the intensity of a previously ignited fire. These actions are taken at a scale equivalent to 12-24 hours of real-time labour, which encapsulates the travel time of the agents to that cell and the time taken by the agents to perform the action.

The agents we've created for this environment can broadly be classified as Heuristic Agents, and Reinforcement Learning Agents. We have also created Randomized Action Agents to serve as a baseline.

1) Heuristic Agents: The initial agent created was a baseline model that used common fire-fighting practices and explicit rules to effectively prevent fire spread. Primarily, to maintain the distance between active fires and agents, we create a 4 unit (200m) dilation around all cells with a fire rank of 1 or greater. Moreover, to determine the dryness of each cell, we get the inverse of the moisture channel $(1-\text{Moisture})$, producing a 90×90 matrix with high values for cells that are dry and far from fires. Finally, the wind vector is accounted for by using the 30×30 heat kernel as a predictor for where the fire is likely to spread in later time steps. To produce the probability distribution shown in the agent channel of Figure 1, these three matrices are multiplied together element-wise and ultimately normalized, from which 3 unique pairs of coordinates are chosen as the agents targets for the day.

2) Reinforcement Learning Agent: This work utilizes the Proximal Policy Optimization (PPO) algorithm (a type of reinforcement learning; [Schulman et al., 2017]) to train an agent. PPO is praised for its stable performance in scenarios with difficult observation and action spaces, matching the challenges of our high-dimensional simulated environment.

The neural network model serving as the core of our agent's decision-making mechanism ingests a compressed representation of the board's state, formatted as a 30x30x8 matrix. The input data is initially processed through a series of convolutional layers, each configured with 3x3 kernels, 16 feature maps, a stride of 2, and no padding, coupled with the Tanh activation function as it was noted to improve performance of the agent [Engstrom et al., 2020]. Subsequently, the extracted features are relayed to a sequence of dense layers, consisting of two layers with 256 neurons each, activated by Tanh, and a final output layer of 6 neurons utilizing the sigmoid activation function. The output layer produces six distinct values, represented as $(x_1, x_2, x_3, y_1, y_2, y_3)$, indicating the optimal positions (x_i, y_i) for each of the three agents.

The simulation is designed to be memory-less - a requirement for Markov Decision Processes - allowing the training of a reinforcement learning agent. Due to the ongoing development of the environment, we have delayed the training of a fully developed RL agent until we have fully grasped an efficient representation of our states into the agent, our actions into the states, and an accurate and powerful reward structure. Currently, our reward function is

$$R_t(S, a) = c_1 * M + c_2 * I + c_3 * D + c_4 * H$$

where:
$M(S, a)$ = Moisture Added to the Environment.
$I(S, a)$ = Decrease in Intensity of Fire (as a result of action taken).
$D(S, a)$ = Distance to edge of the simulation.
$H(S, a)$ = Heuristic with weighted rewards.
and all c_i are hyperparameters.

D. Evaluation Methods

Evaluation of fire spread simulation is a challenge across methodologies. Cell2Fire attempts to mimic Prometheus in the final burn region, though the authors note that one of their major limitations is that they do not model fire suppression actions [Pais et al., 2019] (assumption 4). The authors utilize "structural similarity scores" to identify whether the burn scar region at the end of simulation contains similar local structures. Our simulation was designed such that we could compare the change in dynamics based on agent actions while maintaining visually evaluated fire spread dynamics. As many of the simple interactions in the update step cause feedback to the overall dynamics of the system, and several approximations to those dynamics were made in building the update step, tuning the simulator is a challenging task. We balanced the visual fidelity of fire spread in our region of interest with consistency between runs.

IV. Results and Discussion

To determine the impact agents have on the simulation, multiple different types of agents are evaluated based on their ability to contain fires. Once the final active fire has either been put out or self extinguished, the remaining, untouched surface area is calculated as a percentage of the total surface area referred to as *unburned area*.

Our simulation employs various agents, each utilizing distinct fire-attacking strategies. A simple agent uniformly applies moisture, while a heuristic agent targets locations in the fire's direction of spread. Both apply the same amount of moisture at the same frequency, the only difference is in the locations chosen by the agents. Figure 2 shows the impact of this effective strategy in maximizing the unburned area.

We create 100 initializations of our region of interest and test the strategies of each agent on each scenario. Table I presents the distribution of final unburned area from these tests, showing that the inclusion of a simple agent lead to 410% more fuel being protected than with no agent. The additional strategic placement of moisture by the heuristic agent extends this lead to an additional 45% increase in unburned area over the simple agent by making more efficient use of the resources.

It is important to note that in no cases, with or without the agent, did the unburned area drop below 4%. This is due to spotting behaviour which pushes the fire through an area so quickly that it cannot burn completely, and the disparate islands of unburned cells caused by the fire jumping are surrounded with a border of burned trees protecting them from re-ignition.

Spotting has proven to be particularly problematic in firefighting, as the behaviour is unpredictable and can advance the fire long distances [Martin and Hillen, 2016]. Spotting occurs when embers are blown far from the fire and ignite other areas. Our simulation naturally models this behaviour through the use of our probablistic ignition map update. Figure 3 is a visual inspection of one run of each agent from the same scenario initialization at time = 100. The burn scar is evident from those regions where remaining fuel is zero. The first panel is a control showing the burn scar without active suppression, the simple agent randomly selects from a grid, and the logic of the heuristic agent is outlined in III-C1. In the heuristic agent panel spotting behaviour can be seen as disconnected regions and their resulting burn scar.

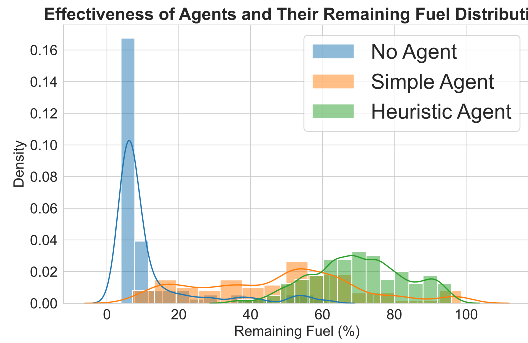

Fig. 2: Histogram and KDE Plot of Forest Integrity after 100 Episodes Per Agent

Agent Type	Mean	Standard Deviation
No Agent	11.76%	12.67%
Simple Agent	48.24%	22.17%
Heuristic Agent	70.15%	12.99%

TABLE I: Distribution of Final Unburned Area

V. Conclusion

This paper has addressed the integration of artificial intelligence in enhancing wildfire management strategies, signifying a paradigm shift from traditional methodologies. By examining the intersection of environmental factors and AI technology, we have outlined the potential of predictive models to revolutionize the ways in which wildfire containment is approached and executed. The implications of our findings extend beyond theoretical modeling, offering practical applications that can be deployed in real-world scenarios. As we move towards the implementation of these AI systems, it is anticipated that wildfire crews will be equipped with more robust tools for decision-making, allowing for a proactive stance in wildfire defense. The goal of this work is to contribute to a body of knowledge that supports the evolution of wildfire management into a more informed, adaptive, and responsive practice.

VI. Limitations / Future Work

Our simulation demonstrates a promising approach to enhancing wildfire prediction accuracy through responsive action. However, it offers opportunities for expansion to improve realism further. Key areas for improvement include the simulation's fidelity—by integrating more complex data layers and behaviors—and enhancing agent realism through greater temporal resolution and diversified interaction methods.

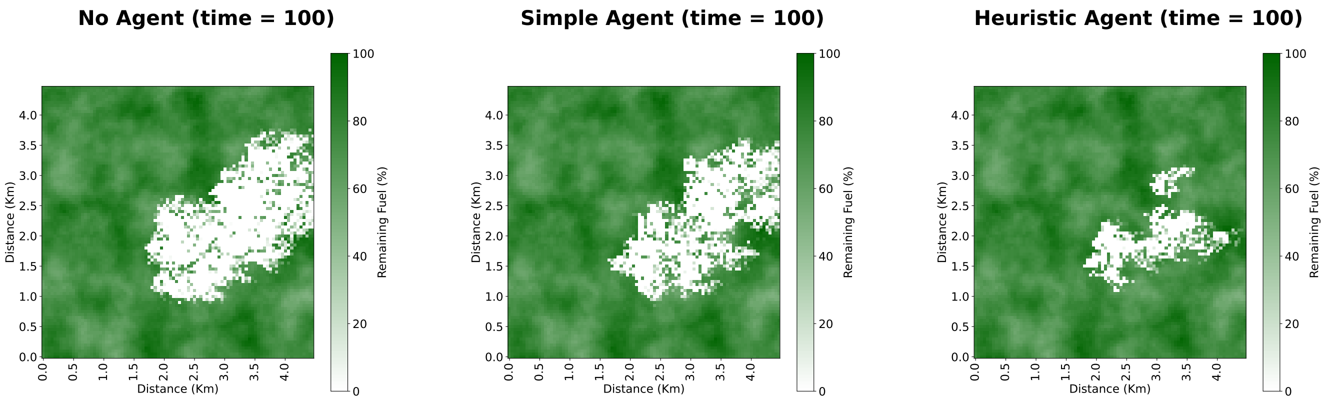

Fig. 3: Spotting Behaviour Across Agent Strategies

A. Simulation Fidelity

Our current model lacks differentiation in fuel types, a critical factor in realistic simulations. Future versions will incorporate fuel types based on specific geographic locations, enhancing the environmental heterogeneity of the simulation.

Presently, wind is simulated with a single unit vector approach that doesn't change during the scenario. An advanced model will include variable wind strengths and temporal fluctuations, using real-world geographical wind data to enrich simulation accuracy.

We recognize the importance elevation plays on the propagation of fire spread. The effect a cell's elevation is relative to the elevation of its neighbors and as a result was a computation that could not be easily parallelized. As a result, it was omitted during development temporarily; incorporating elevation data will improve the simulation's depth and realism.

B. Agent Interactions and Realism

Incorporating satellite images as the starting state will improve realism and allow for ground truth comparisons. This enhancement is aimed at accurately predicting fire propagation in targeted regions.

We aim to increase the temporal resolution of the simulation, allowing for a more continuous and realistic representation of wildfire spread. This involves running multiple environmental steps for each agent action, thereby capturing the dynamic nature of wildfires more accurately.

Expanding the variety of agent types (e.g., aerial units) and their firefighting tactics (e.g., back-burning) will provide a more comprehensive and nuanced understanding of fire management strategies.

Future iterations of our simulation will focus on these areas to create a more sophisticated tool for wildfire prediction and management.

VII. Acknowledgements

We would like to express our appreciation for David Atkinson, researcher in the department of Geography at the University of Victoria. Your insight and knowledge on fire modeling systems was of great help to our team.

References

[BC Government Crews, 2023] BC Government Crews (2023). Wildfire crews. Accessed: 2024-03-08.

[BC Government Summary, 2023] BC Government Summary (2023). Wildfire season summary. Accessed: 2024-03-08.

[Engstrom et al., 2020] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. (2020). Implementation matters in deep policy gradients: A case study on ppo and trpo.

[Farguell et al., 2019] Farguell, A., Cortés, A., Margalef, T., Miró, J. R., and Mercader, J. (2019). Scalability of a multi-physics system for forest fire spread prediction in multi-core platforms. *The Journal of Supercomputing*, 75:1163–1174.

[Fire manual, 2016] Fire manual (2016). *S-241 (BC) Fire Assessment TRAINEE MANUAL*.

[Kourtz et al., 1977] Kourtz, P., Nozaki, S., and O"Regan, W. (1977). *Forest fires in the computer – a model to predict the perimeter location of a forest fire*. Fisheries and Environment Canada.

[Martin and Hillen, 2016] Martin, J. and Hillen, T. (2016). The spotting distribution of wildfires. *Applied Sciences*, 6(6).

[Pais et al., 2019] Pais, C., Carrasco, J., Martell, D. L., Weintraub, A., and Woodruff, D. L. (2019). Cell2fire: A cell-based forest fire growth model to support strategic landscape management planning. In *Frontiers in Forests and Global Change*.

[Reid et al., 2016] Reid, C. E., Brauer, M., Johnston, F. H., Jerrett, M., Balmes, J. R., and Elliott, C. T. (2016). Critical review of health impacts of wildfire smoke exposure. *Environmental Health Perspectives*, 124:1334 – 1343.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

[Tymstra et al., 2010] Tymstra, C., Bryce, R., Wotton, B., Taylor, S., and Armitage, O. (2010). *Development and Structure of prometheus: the Canadian Wildland fire Growth Simulation Model*. Nat. Resour. Can.