Combining MCTS and A3C for Prediction of Spatially Spreading Processes in Forest Wildfire Settings

Sriram Ganapathi Subramanian(✉) and Mark Crowley

Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Canada
{s2ganapa,mcrowley}@uwaterloo.ca

Abstract. In recent years, Deep Reinforcement Learning (RL) algorithms have shown super-human performance in a variety Atari and classic board games like chess and GO. Research into applications of RL in other domains with spatial considerations like environmental planning are still in their nascent stages. In this paper, we introduce a novel combination of Monte-Carlo Tree Search (MCTS) and A3C algorithms on an online simulator of a wildfire, on a pair of forest fires in Northern Alberta (Fort McMurray and Richardson fires) and on historical Saskatchewan fires previously compared by others to a physics-based simulator. We conduct several experiments to predict fire spread for several days before and after the given spatial information of fire spread and ignition points. Our results show that the advancements in Deep RL applications in the gaming world have advantages in spatially spreading real-world problems like forest fires.

Keywords: Monte-Carlo Tree Search · A3C
Reinforcement Learning · Forest wildfire
Computational sustainability

1 Introduction

Recent advances in **Deep Reinforcement Learning (RL)** such as AlphaGo [1] and the more recent Alpha Zero [2] algorithms show that it is possible to combine **Monte Carlo Tree Search (MCTS)** intelligently with Neural Networks to get a sophisticated system that can beat the best Go players in the world. This serves as a motivation to try similar approaches on spatially spreading real world phenomena such as forest wildfires. Forest fires present a more difficult challenge than these board games as the dynamics of the environment in the real world are more prone to uncertainty and randomness. Making learning agents that can give highly accurate solutions in such domains has been attempted before on simpler problems [3] but stopped at the simulation level as a demonstration. This paper introduces a novel algorithmic framework for this problem, **MCTS-A3C**, which merges two existing Deep RL approaches. To validate our approach we use data from real and simulated forest fire events:

E. Bagheri and J. C. K. Cheung (Eds.): Canadian AI 2018, LNAI 10832, pp. 285–291, 2018.
https://doi.org/10.1007/978-3-319-89656-4_28

Simulated wildfires: For clean images and as a faster testing domain, we are using the forest wildfire simulator by Nova online [4]. Figure 1(d) shows the simulator at work. We used a variety of environment variable settings to create a realistic set of simulations. The fire spread with the given conditions is trained and tested with our RL algorithm.

Alberta wildfires: In an earlier work [5] we introduced the idea of using RL for prediction in spatiotemporally spreading domains such as forest wildfires. For validation we used satellite images from two forest fires in Northern Alberta (Fig. 1(b) and (c)) and compared a number of classical RL algorithms as well as the standard Deep RL algorithm A3C [6] for this domain. We consider this data again here and show that our new MCTS-A3C algorithm outperforms all these.

Saskatchewan wildfires: Physics-based simulations are the current state of the art in the wildfire analysis literature. So, a comparative study will also be made to demonstrate the performance of the MCTS-A3C algorithm against the physics-based simulation in Burn-P3 (BP3) [7] used for analysis of historical fires in central Saskatchewan (52.9399°N, 106.4509°W). For information on the input/output formats and methodology for BP3 please see [7].

2 Problem Formulation

We define a Markov Decision Process (MDP) $< S, A, P, R >$ where the set of states S describes any location on the landscape and where the 'agent' taking actions is a fire spreading across the landscape. A state $s \in S$ corresponds to the state of a cell in the landscape $(x, y, te, l, w, d, rh, r, i, tm)$ where x, y are the location of the cell, te is the temperature at the particular time and location and l is the land cover type of the cell derived from satellite images (values include: water, vegetation, built up, bare land, other), w is wind speed, d is wind direction, rh is relative humidity, i is the intensity of fire as defined in [8], tm is the time considered in days from the start of fire, and r is the average amount of the rainfall at the spatial coordinates during the time of study. These variables are the largest contributing factors to fire spread in the Canadian Forest fire weather index [9]. The action $a \in A$ indicates the direction the fire at a particular cell 'chooses' to move: North, North-West, North-East, South, South-West, South-East, East or West or to stay put (see Fig. 1(a)). The dynamics function for any cell $P(s'|s, a)$ is a mapping from one state s to the next s' given an action a. The reward function R maps a cell state to a continuous value in the range $[-1, 1]$ and is based on the land cover. **The goal is to learn a policy for this agent that recreates the spread of the fire observed in later satellite or simulated images by maximizing discounted rewards designed to encourage high accuracy spread prediction.**

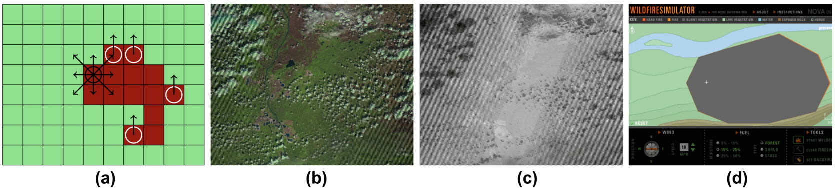

Fig. 1. Forest Wildfire Satellite Data Domain: (a) A schematic of the wildfire motion domain at a particular state and timestep. (b) Raw color satellite image of a target area. (c) Thermal image of the same area at (b) showing hotspots on fire (dark). (d) Nova fire simulator (Color figure online)

3 The MCTS-A3C Algorithm

The high level pseudo-code is given in Algorithm 1. We use an MCTS [10] search tree that contains an average reward value (X) and a visit count (N) for all the nodes. Each node is a cell on fire and has its own state space S. The fire is made to start at the ignition points and each node of the search tree is comprised of a cell burning in the resolution of the source data (30 m in case of Alberta fires and 300 m in case of Saskatchewan fires). The selection step in the algorithm is given by the UCT strategy [10]. To encourage exploration, random selection is also done with probability ϵ. The root node of the tree contains the ignition point and an action choice (selection step) leads to another cell being on fire and this results in child nodes containing the new ignited cell on fire. This process continues until we reach end of the tree containing all the visited nodes. In the next expansion phase an unvisited node is randomly added to the search tree and we simulate forward using the A3C algorithm [6]. The initial state for A3C has 84×84 cells ([11]) surrounding the center ignited cell as its start state. Each worker of A3C is defined as an instance of fire with a distinct environment and related networks. The algorithm spins off a separate worker on a distinct thread every time a new flame is started in the search space. Then each worker propagates fire in its own local condition, obtains rewards and updates the

Algorithm 1. MCTS-A3C

```
procedure MCTS (ROOTNODE, TIME)
    while time=true do
        currentnode ← rootnode
        while currentnode!=lastnode do
            lastnode ← currentnode
            currentnode ← SELECTION(currentnode)
        end while
        lastnode ← EXPANSION(lastnode)
        SIMULATE using A3C
        while currentnode do
            Backpropogate(currentnode)
            currentnode ← parent(currentnode)
        end while
    end while
end procedure
```

global network. Next, the back propagation step increments the visit count and updates the reward values of each node. The algorithm is configured to stop after the desired number of days in the input state (t) comes to an end for the experiment of consideration. If the intensity of fire at a cell falls below 0.3, it is empirically determined to cease to burn and is stopped from spreading further (in the case of A3C) and hence is removed from the search tree (in MCTS) until it is reignited. The model architecture for A3C follows the scheme in [11].

4 Experimental Setup

The first set of experiments (A)–(F) use the Alberta forest fire datasets. In experiment (A) we consider three consecutive, evenly spaced, timesteps and we want to test the ability of the algorithms to predict the middle time step. The ignition points for training come from the time step 1 and the training happens using the information in time step 3. This is used to determine the fire spread at the intermediate time step 2.

Experiment (B) starts with the initial state of the fire, providing rewards based on time step 2 and the algorithm predicts fire locations at time step 3. This experiment is similar to asking the question: Where will the fire spread in the next 16 days, given its position currently?

In experiments (C)–(F) we apply the learned policy from the Richardson fire to the Fort McMurray fire over four 16-day time steps in the Fort McMurray data after giving the ignition points as the input. This was a fire that happened in Northern Alberta, 5 years after the Richardson fire used for training. As the regions are similar and very near each other, the general properties that the model encapsulates should remain relevant. We provide the start state (satellite image corresponding to the start date of the Richardson Fire, for experiments (A, B) and Fort McMurray fire for experiments(C–F)) to determine the spatial and landscape features of the region under consideration.

In experiment (G), we use the Nova simulator to generate wildfire data and predict forward based on the given situation. Experiment (H) uses the Saskatchewan fire data from 1981 to 1992 for training the fire spread model and analyzed for the years from 1993 to 2002. We also compare to existing results using burn probabilities for 1993 from [7]. In the same way, Experiment (I) uses data from 1981–2002 and tests for 2003–2008. This is done to test the performance on an experiment having more training samples and less predicted spread. For both experiments (H) and (I), the RL algorithms and BP3 were run on 300 m resolution cells as this was the lowest resolution of data source grids for the cell based inputs in the experiments carried out in [7], and due to limitations in computation time.

In our experiment, the results from BP3 for every cell were translated into binary with the last three ranges characterized as a burn and the first four ranges characterized as a no-burn. We have compared the performance of our combined MCTS-A3C algorithm to the A3C, MCTS, DQN algorithms [11] and DQN with prioritized experience relay (PER) [12]. The discount rate γ was 0.99 as the goal is to make the agent strive for a long term reward and the exploration for the ϵ greedy policy was kept as 0.05, which is along the lines of exploration rates in previous works ([6,11]). We have rescaled every input to grey scale with 84×84 pixels as followed in [11]. We have 25 input frames per second of simulation in all the experiments and thus, an hour of training yields 90000 observations. For all the experiments, to determine if a cell was on fire, a threshold value beyond which a cell must burn was determined for the value function to balance true positives and false positives. Results are compared against the corresponding satellite images for accuracy.

5 Results

Referring to Table 1 we can notice some general trends in the accuracies for all the experiments from (A) to (I). The MCTS-A3C algorithm beats all the remaining algorithms in terms of the accuracy. The general rule seems to be MCTS-A3C > A3C > DQN with PER > DQN > MCTS. The deep learning based approaches perform better than MCTS in most experiments. MCTS has advantages in some experiments such as (B) as it is able to fit the experiments with a limited data set that predicts forward in time better than other algorithms. A3C on the other hand is able to do well in experiments that predict in the middle of a time step (like (A)) and also in experiments that transfer the learned policy to a new test domain ((C)). Thus, as expected, MCTS-A3C, which is a combination of these two algorithms, does well in all test domains. Also, as a general rule the accuracies for all experiments decrease from (C) to (F). This is because as we keep moving into the fire season the fire becomes progressively more intense and erratic, which makes prediction tougher. These results show the generalizability of an RL approach to prediction of a spatially spreading process like fire.

For the simulated fire experiment (G), we see that MCTS-A3C has a clear advantage over all the other algorithms. In the experiments related to the

Table 1. Average Accuracy for each algorithm on the different test scenarios for Alberta fires.

Experiment	MCTS	A3C	DQN	DQN(PER)	MCTS-A3C
A	61.3%	87.3%	73.2%	79.4%	92.4%
B	60.2%	53.2%	49.5%	51.8%	90.8%
C	65.3%	90.1%	49.7%	50.8%	90.3%
D	55.7%	81.8%	72.3%	75.9%	73.4%
E	49.7%	50.8%	45.6%	48.5%	57.8%
F	5.8%	13.4%	9.4%	10.8%	50.6%
G	80.7%	84.2%	81.5%	82.7%	95.8%
H	51.3%	58.5%	52.1%	53.4%	65.8%
I	60.1%	67.2%	60.5%	62.7%	73.8%

Table 2. Accuracy values for the different ecoregions in the area of Saskatchewan for the experiment (H) and (I). Only the MCTS-A3C algorithm is considered for this comparison and is denoted as RL.

EcoRegion	BP3 (H)	MCTS-A3C (H)	BP3 (I)	MCTS-A3C (I)
Boreal Transition	33.7%	68.4%	36.7%	69.3%
Mid-boreal Lowland	53.5%	49.1%	56.5%	69.1%
Mid-boreal Upland	59.2%	75.0%	57.5%	79.2%
Churchill river upland	62.1%	70.5%	66.3%	77.8%

Saskatchewan fires seen in (H) and (I) we can see that all the algorithms have lower average accuracy. This is because there is a lot of inaccuracies in the source data [7] for experiments (H) and (I) and also because we compare several years forward from the training data set for the experiment (H). As more data samples are available for training, performance get better in the experiment (I) as compared to (H). The performance of the MCTS-A3C algorithm in the experiments (G), (H) and (I) demonstrates that it is scalable. The accuracies we obtained for the Saskatchewan fire (Table 2) using MCTS-A3C correspond well with detailed landscape burn proportions using BP3 reported in [7]. Our algorithm has a higher overall average accuracy in comparison to the BP3 outputs and also it does better in most eco regions. The Mid-boreal Lowland region for experiment (H) is the only region in which the BP3 outperforms our RL algorithm. This region is highlighted in [7] as having a much higher uncertainty due to large inaccuracies in the source fuel data. This shows that MCTS-A3C falls short of the physics based simulator only when the source data sets are affected too much by distortions and noise. But even then we can see that in experiment (I) MCTS-A3C outperforms the BP3 simulator in the same region. In general we do better in (I) as compared to (H). This shows that given enough data samples an RL approach can do better than physics based simulators. This is a clear advantage as ever larger datasets and better hyper-spectral sensors and drone images become available during forest wildfires as compared to the last couple of decades.

Concluding Remark: We have introduced and demonstrated the effectiveness of the MCTS-A3C algorithm for spatial prediction on a range of simulated and historical forest fire data. An advantage of this method is that we don't need to use expert rules and it generalizes and scales well. In future, we plan to carry out experiments on other spatial domains and further extend this algorithm to broader classes of RL problems.

References

1. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
2. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of go without human knowledge. Nature **550**(7676), 354–359 (2017)
3. Forsell, N., Garcia, F., Sabbadin, R.: Reinforcement learning for spatial processes. In: 18th World IMACS/MODSIM Congress, Cairns, Australia, pp. 755–761 (2009)
4. Groleau, R.: Fire wars. http://www.pbs.org/wgbh/nova/fire/simulation.html. Accessed 30 Nov 2017
5. Ganapathi Subramanian, S., Crowley, M.: Learning forest wildfire dynamics from satellite images using reinforcement learning. In: Conference on Reinforcement Learning and Decision Making, Ann Arbor, MI, USA (2017)
6. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
7. Parisien, M.A., Kafka, V., Hirsch, K., Todd, J., Lavoie, S., Maczek, P., et al.: Mapping Wildfire Susceptibility With the Burn-p3 Simulation Model. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre Edmonton (AB) (2005)
8. Keeley, J.E.: Fire intensity, fire severity and burn severity: a brief review and suggested usage. Int. J. Wildland Fire **18**(1), 116–126 (2009)
9. Cortez, P., Morais, A.: A data mining approach to predict forest fires using meteorological data. In: Proceedings of the 13th Portugese conference on Artificial Intelligence, December, Guimares, Portugal, 512–523 (2007)
10. Kocsis, L., Szepesvári, C.: Bandit based monte-carlo planning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 282–293. Springer, Heidelberg (2006). https://doi.org/10.1007/11871842_29
11. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
12. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv preprint abs/1511.05952 (2015)