Vulcano: Operational Fire Suppression Management Using Deep Reinforcement Learning

Extended Abstract

Cristobal Pais
University of California Berkeley, Industrial Engineering and Operations Research
cpaismz@berkeley.edu

ABSTRACT

Vulcano is a fire-management system based on deep reinforcement learning (DRL). Using simulated trajectories from a state-of-the-art simulator, agents are trained to select areas that should be treated to minimize fire propagation. We focus on the operational problem where fire suppression teams are deployed after detecting an ignition and collaborative strategies are critical to contain the fire. We propose a new algorithm based on centralized training with decentralized execution, modifying the reward and advantage functions to provide each agent with critical information about the teams. Experiments demonstrate the performance of the method compared to traditional approaches.

KEYWORDS

Wildfire management, Multi-agent, Deep reinforcement learning

ACM Reference Format:
Cristobal Pais. 2020. Vulcano: Operational Fire Suppression Management Using Deep Reinforcement Learning. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020*, IFAAMAS, 3 pages.

1 INTRODUCTION

In recent years, we have seen increasing burned areas by forest fires worldwide due to changes in temperature and precipitation levels [21, 26, 27]. Wildfires have consumed important areas and forest resources, as a result, fire management expenditures have increased and thousands of homes and many lives have been lost. Although DRL algorithms have erupted in several fields and applications [15], their implementation in fire management problems have only been discussed [28] and only scarce learning models can be found in the literature [1, 3, 20]. To date, several efforts have been done to integrate fire-management with real/simulated data [23–25], however, these approaches have strong limitations such as poor scalability when including uncertainties and a fire-expert dependent performance, given the complexity of the problem.

To deal with the intractability of the problem as the number of agents increase when applying centralized methods, we focus on a *centralized training and decentralized execution* approach [7, 10, 12, 13], where agents have access to the true state of the system during training, using this information to boost their performance during the execution phase. We propose a local reward extension to a multi-agent deep reinforcement learning (MADRL) actor-critic algorithm (COMA) [8] to find efficient collaboration strategies for subsets of agents in the context of fire suppression planning, modifying the landscape to minimize the fire propagation. A novel training environment is built on the top of a fire growth simulator [18]. Experiments in real forests show the potential of the system.

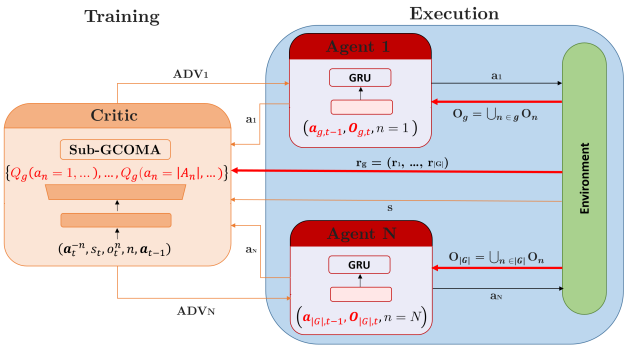

Figure 1: Sub-Groups COMA structure. A central critic calculates the counterfactual advantage function to update agents' policies. Q-functions values are estimated for each sub-group g.

2 METHODS

Environment. We consider a multi-agent fully cooperative world [16, 19] with $n \in N$ agents, a set of actions $a_t^n \in A_n$ for each time-step $t \in T$, and a set of observations O_n. The true state of the environment is described by $s \in S$. Each agent selects an action a_t^n at time-step t by following a stochastic policy $\pi_n : A_n \times O_n \rightarrow [0, 1]$. A joint action vector $\boldsymbol{a}_t = \left(a_t^1, ..., a_t^n\right) \in \boldsymbol{A} = A_1 \times ... \times A_n$ is given to the environment, defining a state transition function $P(s_{t+1}|s_t, \boldsymbol{a}_t) : S \times \boldsymbol{A} \times S \rightarrow [0, 1]$. Agents can perform six different actions – four directional movements, harvest, and wait/rest – in a forest mapped into a grid composed of cells with an identical area. Fire is simulated by tracking the state of all cells as the fire progresses through discrete time steps. A state is composed by (i) the expected fire progress, (ii) number of actions needed to harvest a cell, (iii) topographic information, (iv) the agents' positions, and (v) the weather forecast. They are penalized by a proportional factor to the fire progress, the number of burned cells by the end of the episode (-1), and being caught by fire (-100). Available cells are rewarded (+2) at the end of the episode.

Training Architecture We extend COMA [8] with the concept of local rewards and Q-functions for each sub-group of agents $g \in G$ in the context of MARL [2]. In our model, each agent

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

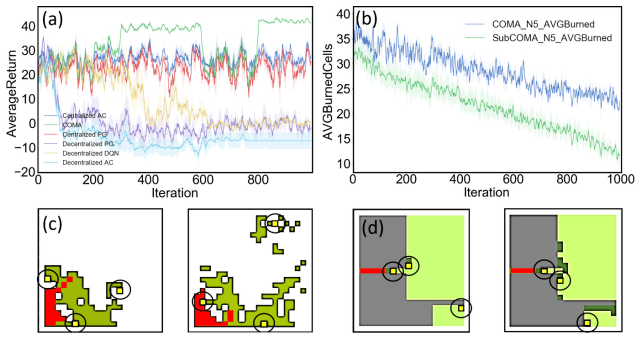

Figure 2: Results samples. (a) Homogeneous instance training comparison ($n = 5$, 1600 cells). (b) COMA and SubG-COMA performance comparison in a heterogeneous landscape ($n = 5$). (c) Visualization of SubG-COMA and COMA policies. Agents are highlighted by black circles, harvested cells in dark green, and fire in brown.

n belongs to a unique sub-group g; $\boldsymbol{a}_g$ is the vector of actions taken by the agents in g; $r_g(s, \boldsymbol{a}_g)$ the local-reward obtained by g when taking actions $\boldsymbol{a}_g$ on state s; and $Q_g(s, \boldsymbol{a}_g)$ the sub-group action-state function (Figure 1). A central critic calculates the advantage function for an agent n from the sub-group g as $A^n(s, \boldsymbol{a}_g) = Q_g(s, \boldsymbol{a}_g) - \sum_{a_n'} \pi^n(a_n'|\tau_n) Q_g(s, (\boldsymbol{a}_g^{-n}, a_n'))$.

Actors are represented by gated recurrent units (GRUs) [4] using fully connected layers for processing the inputs and outputs from the hidden state, and the centralized critic is defined with 5 fully-connected layers of 64 units with ReLu activation. The optimal learning rate (0.0005), discount factor γ (0.99), and batch size (64) are found using derivative-free optimization methods [5, 6].

Experiments. Policy gradient/Actor-Critic (PG/AC) [9, 22], Double Deep Q-Networks (DDQN) [11], and Hysteretic Q-learning (HQL) [14, 17] algorithms are implemented in their centralized and decentralized versions for benchmarking and exploring the most suitable approach for the environment. We compare the average return of all the implemented methods, as well as the average number of burned cells by the end of an episode. Models are trained for 100,000 episodes, averaging metrics every 100 episodes. We vary the number of agents $n \in \{1, ..., 5\}$.

Two experimental sets are used to assess the performance of the algorithms. The first set consists of real Canadian landscapes where fires on homogeneous and heterogeneous (multiple fuel types and non-flammable cells) forests of different sizes are simulated. The second set contains five generated landscapes to assess the performance of the algorithms in specific coordination tasks. Weather conditions and ignition probabilities are based on historical datasets[1]. The open-source implementation and detailed results can be found in http://github.com/cpaismz89/Vulcano_DRL.

3 RESULTS

Homogeneous. We solve a 9, 400, and 1600 cells square instances. All algorithms obtain similar performance in the first and second instances, however, agents tend to require more time (+50%) to contain the fire when no centralized training is performed due to the lack of collaboration strategies. In addition, returns variance is significantly increased (+20%) with respect to the number of agents when using decentralized methods and are easily outperformed by centralized ones when dealing with larger homogeneous forests (e.g., 40×40, Figure 2 (a)) where coordination is critical. Centralized methods tend to dominate in terms of average return (30% less of average area burned) but tend to be noisy. On average, COMA/SubG-COMA converge faster (requiring one-third of iterations) and are stable, presenting a similar performance for a different number of subgroups. Q-Learning methods such as DDQN and HQL did not reach good performance and were dominated by the rest of the algorithms.

Heterogeneous. COMA and SubG-COMA are able to learn high-quality policies faster and with less variance than other algorithms as we increase the number of agents (Figure 2 (b)). The performance of decentralized algorithms is worse than in the homogeneous case since the fire dynamic is affected by the forest structure and coordination becomes crucial to contain the fire. Centralized algorithms are still competitive but become intractable after increasing the number of agents beyond five. We observe that COMA agents tend to over-harvest the forest in comparison to SubG-COMA as the number of agents is increased. The explanation is that COMA agents receive a noisy approximation of their contribution to the global reward, not capturing their real impact, thus, performing sub-optimal actions. For example, we observe the agents trained by SubG-COMA and COMA on the 20×20 instance (Figure 2 (c)) where agents 1 and 2 are located at the bottom-center of the landscape and agent 3 is placed on the north-east side. SubG-COMA agents find an efficient collaborative strategy, using $|G| = 2$ by creating a sub-group with agents 1 and 2. This happens because the third agent observes a different reward function, allowing it to understand that harvesting cells on the north-east side is not useful to contain the fire.

Challenges. Comparing the performance of our extension and COMA (Figures 2 (d)), we see how agents following SubG-COMA are able to find subtle but more efficient/complex collaborative strategies. On the left side, SubG-COMA agents discover that harvesting next to the fire is enough to protect the land beyond the mountains section (gray cells) while COMA agents continue to harvest cells in non-critical places. We observe this on the isolated agent where its optimal action is to *wait* since it cannot help to contain the fire, however, the agent tends to harvest cells due to a failed credit assignment when training the agents.

4 CONCLUSIONS

We tested state-of-the-art MADRL algorithms in a novel fire suppression environment. An extension of a centralized training and decentralized execution AC algorithm with local rewards and Q-functions for sub-groups was implemented, outperforming traditional algorithms in a cooperative setting. In order to exploit the SubG-COMA extension, sub-groups should be carefully selected to exploit complex interactions within teams, matching agents with significant collaboration. Our results represent a novel DRL application on fire suppression planning with the potential of multiple extensions and real-life applications.

[1] http://www.firegrowthmodel.ca/prometheus/software_e.php

REFERENCES

[1] Paolo Avesani, Anna Perini, and Francesco Ricci. 2000. Interactive case-based planning for forest fire management. *Applied Intelligence* 13, 1 (2000), 41–57.

[2] Drew Bagnell and Andrew Y Ng. 2006. On local rewards and scaling distributed reinforcement learning. In *Advances in Neural Information Processing Systems*. 91–98.

[3] Christopher Bone and Suzana Dragićević. 2010. Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management. *Computers, Environment and Urban Systems* 34, 2 (2010), 162–174.

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[5] Andrew R Conn, Katya Scheinberg, and Ph L Toint. 1997. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical programming* 79, 1-3 (1997), 397.

[6] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. 2009. *Introduction to derivative-free optimization*. Vol. 8. Siam.

[7] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*. 2137–2145.

[8] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.

[9] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research* 5, Nov (2004), 1471–1530.

[10] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 66–83.

[11] Hado V Hasselt. 2010. Double Q-learning. In *Advances in neural information processing systems*. 2613–2621.

[12] Emilio Jorge, Mikael Kågebäck, Fredrik D Johansson, and Emil Gustavsson. 2016. Learning to play guess who? and inventing a grounded language as a consequence. *arXiv preprint arXiv:1611.03218* (2016).

[13] Landon Kraemer and Bikramjit Banerjee. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* 190 (2016), 82–94.

[14] Martin Lauer and Martin Riedmiller. 2000. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer.

[15] Yuxi Li. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274* (2017).

[16] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier, 157–163.

[17] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. 2007. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 64–69.

[18] Cristobal Pais, Jaime Carrasco, David L Martell, Andres Weintraub, and David L Woodruff. 2019. Cell2Fire: A Cell Based Forest Fire Growth Model. *arXiv preprint arXiv:1905.09317* (2019).

[19] Liviu Panait and Sean Luke. 2005. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems* 11, 3 (2005), 387–434.

[20] Francesco Ricci, Paolo Avesani, and Anna Perini. 1999. Cases on fire: applying CBR to emergency management. *NEW REV APPL EXPERT SYS* 5 (1999), 175–190.

[21] Steven W Running. 2006. Is global warming causing more, larger wildfires? *Science* 313, 5789 (2006), 927–928.

[22] Richard S Sutton and Andrew G Barto. 2011. Reinforcement learning: An introduction. (2011).

[23] Yu Wei. 2012. Optimize landscape fuel treatment locations to create control opportunities for future fires. *Canadian Journal of Forest Research* 42, 6 (2012), 1002–1014.

[24] Yu Wei, Douglas Rideout, and Andy Kirsch. 2008. An optimization model for locating fuel treatments across a landscape to reduce expected fire losses. *Canadian Journal of Forest Research* 38, 4 (2008), 868–877.

[25] Yu Wei, Douglas B Rideout, and Thomas B Hall. 2011. Toward efficient management of large fires: a mixed integer programming model and two iterative approaches. *Forest Science* 57, 5 (2011), 435–447.

[26] AL Westerling, HG Hidalgo, and DR Cayan. [n. d.]. tW Swetnam. 2006. *Warming and Earlier Spring Increase Western US Forest Wildfire Activity. Science* 313 ([n. d.]).

[27] Anthony LeRoy Westerling. 2016. Increasing western US forest wildfire activity: sensitivity to changes in the timing of spring. *Philosophical Transactions of the Royal Society B: Biological Sciences* 371, 1696 (2016), 20150178.

[28] Marco A Wiering and Marco Dorigo. 1998. Learning to control forest fires. (1998).