Neural Tree Expansion for Prioritized Multi-Agent Path Deconfliction

Khushal Brahmbhatt
Oregon State University
Corvallis, OR
brahmbhk@oregonstate.edu

Andrew Festa
Oregon State University
Corvallis, OR
festaa@oregonstate.edu

Kavinda Senewiratne
Oregon State University
Corvallis, OR
senewiry@oregonstate.edu

KEYWORDS

multi-agent, decentralized, Monte-Carlo tree search, priority path planning

1 INTRODUCTION

Due to rising global temperatures, there has been a significant increase in the number of wildland fires in the USA. In a span of 30 years, the number of annual wildfires have doubled from approximately 500 to 1000 [2]. There were about 1.3 million acres burned by wildland fires in 1983, and thhis number grew to approximately 10 million acres by 2020 [3]. In 2021, the Bootleg Fire which started in Southern Oregon, burned almost 500,000 acres of land [1]. This is almost half the number of acres totally burned in 1983 across the entire USA. In 2018, the estimated insured losses totaled to about \$10Bn in the USA, and the Federal Government spent \$3Bn for wildfire suppression. More importantly, there were countless lives (civilian and firefighters) lost [4]. If we can better fight these fires and predict how they are going to evolve over time, these major losses may be able to be mitigated.

Fighting these fires requires fast decision making under extreme conditions, and a single, wrong action could result in losing lives or damaging valuable assets. Data and information are vital for any decision making process, and fighting wildland fires requires a lot of real-time information as there are many constantly changing factors (e.g. weather conditions, fuel type, fuel moisture, etc), which affect the rate of spread. Specifically, observing how the boundaries of the fire are growing provides a lot of information useful for predicting the future path of the fires. These boundaries, called fire fronts, are where the fires are actively burning, and any given wildland fire usually has several fire fronts. Thus, monitoring these boundaries is a difficult task due to the highly dynamic and time-sensitive nature of the data collected.

There are many methods used to monitor the behavior of the fire and conditions around it. A primary method is satellite imagery. Generally, it is used for initially detecting a wildland fire. However, due to the low temporal resolution of the imagery they provide, it is difficult to rely solely on this data as the fronts are constantly changing. Stationary, terrestrial sensors could also be used. However, in a wildland fire, communication is extremely limited. This could be due to infrastructure that has been damaged by the fire or that the fire is taking place is an extremely remote location where there is no ground communication infrastructure in place.

All of this amounts to needing a real-time monitoring solution that is capable of operating in areas of low visibility without relying on communication to a centralized location. Unmanned Aerial Vehicles (UAVs) are ideal for the task of reconnaissance, but are limited in air time due to the current limits of battery technology. Due to this, any autonomous system used, would need to plan efficiently to make the most out of the resources available to them. Using a large number of UAVs would allow the task to be spread over several agents and thus help alleviate the battery limitation. Drone swarms have been proposed in helping with monitoring and suppressing wildland fires [5]. However, as the size of the swarm increases and the communication remains tenuous, it quickly becomes intractable for the drones to plan non-conflicting paths.

This work contributes to the problem of finding near-optimal prioritized non-conflicting paths among multiple agents with limited communication by defining an explicit objective-based prioritization scheme that can be applied to the work outlined in [10]. In this way, different aspect of a meta-problem can be addressed with high importance even when the number of agents and objectives in the system becomes intractable for directly computing an optimal solution.

We use a Neural Tree Expansion (NTE) algorithm [10] that leverages Monte-Carlo Tree Search and policy and value networks to learn optimal paths for each drone from a starting position to a given goal position on the fire front. To plan for collision-free paths between the drones, we need a way to deconflict colliding paths. To this end, we use a priority-based reward function to determine which drone gets priority along its planned path and which one has to wait or find a different path. The priorities are assigned to the goal positions and change with time to represent the criticality of that part of the fire front. This priority is used in a shared system reward that is passed to each NTE model associated with locally planning each drone's trajectory.

2 BACKGROUND

In multi-agent path planning, multiple agents attempt to find the shortest paths to reach their individual goals. However, in order to avoid colliding, each agent may have to follow a sub-optimal path compared to if they were the only agent in the system. The problem thus becomes how to search a set of non-conflicting paths in order to jointly minimize the time each agent takes to reach its goal and the total throughput of all the agents in the system. To add to the challenge of the problem, there often exists a priority among the agents. That is, it would be desired that a certain agent, or subset of agents, reach their targets faster than those with a lower priority. At the extreme ends, solving this optimization problem is often done in one of two ways: centralized and decentralized. In a completely centralized approach, each agent is able to communicate with the other agents in the system, while in a completely decentralized solution, each agent only performs its own computations or a local search without additional knowledge from other components in

ROB 537, , October 2021, Corvallis, OR, USA. © Learning-Based Controls

the system. There is an inherent trade-off in the two approaches in terms of optimality and resource cost, and so most approaches fall somewhere in between [15].

The centralized approach is able to compute the optimal solution that would jointly minimize the multiple objectives, but it comes at the cost of intractability in the number of agents in the system [6]. In fact, this problem has been shown to be NP-hard [8] and thus cannot be reliable computed for any scenario with a sizeable number of agents. Alternatively, in a decentralized approach, each agent only searches a local space it has not global information regarding the other agents in the system. The issue comes in that multiple agents may make local decisions which conflict with the local decisions of other nearby agents.

2.1 Prior Work

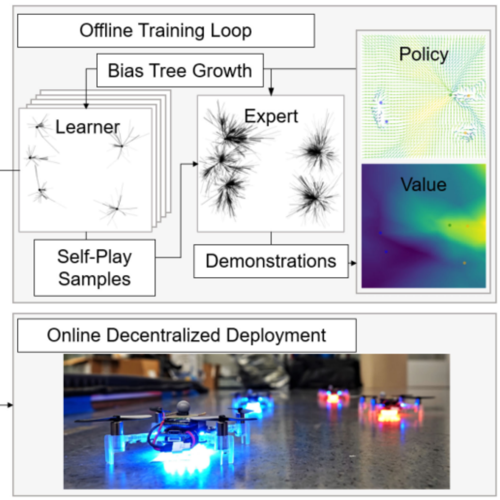
In order to combat the intractability inherent in computing a global, optimal solution, many approaches construct heuristics that seek to efficiently find near-optimal paths. A*, and variations on it, have been shown to yield decent results [12, 14], and conflict-based search (CBS) [11] is widely used and extended in order to cut down on the number of path conflicts that must be searched through. The general idea that CBS introduces is searching at two levels: a global tree with lower resolution and a low level search for each agent. The high-level search is meant to find the points in a path where the agents may collide. The agents themselves are responsible for searching locally in order to resolve those path conflicts. In doing this, the total system is able to search only over the parts of the paths that give rise to the conflicts, and thus it greatly cuts down on the search space. However, this approach falls short when there may be multiple objectives or each agent may have a unique objective [9]. Several attempts have been made toward this particular challenge, [6, 9, 13], but they fail to consider the situation where a priority may exist between the agents in the systems. Instead, they seek to minimize the total time it takes for all of the agents to reach their objectives rather than a weighted average for each individual agent, which is a necessary consideration in many situations, as discussed previously. In the case of UAVs tracking wildfire spread, the agents may not all be able to reliably communicate and different points along the boundary may be more critical due to proximity to urban centers. Thus, a fully centralized approach is not suitable, and a prioritization scheme is necessary in order to ensure that critical parts of the overall problem are addressed first.

Some work has been proposed to address this specific gap. In [16], the authors use a prioritization heuristic based on the number of path choices available to an agent to deconflict paths. In [7], the path planning problem is formulated as a mixed integer problem defined using temporal logic and use a search optimization method to find the optimal collision-free paths based on a fairness function. However, these rely on knowing the robot's path options upfront and assign the priority based on characteristics of the agent and environment rather than the task of the agent(s).

2.2 Neural Tree Expansion

The approach in [10] uses dual a Monte-Carlo tree search to both search the global space and the local space of each agent. The global search, depicted as the Expert in figure 1, is only used for training each local agent and is not used when the agents are deployed.

Figure 1: Neural Tree Expansion Training Loop



This work extends the idea proposed in CBS of searching along conflicting paths to apply the sequential search to the agents rather than the paths, and it outlines two core loops, as depicted in figure 1: an offline training loop, where each agent is trained to imitate an expert, and an online decentralized deployment loop, where the trained agents are deployed to the real world. The training loop contains both the learning agents, all of which only have partial information, and an expert oracle, that has complete and global information. Both the learning agents and the oracle use Monte Carlo Tree Search (MCTS) to explore the possible local paths to a particular depth and is further biased by a value function (during the selection step of MCTS) and a policy function (during the expansion step of MCTS). The learners explore various states (which are all labeled by the expert) and are then used to train the policy and value networks. This loop is repeated until the networks have converged or achieved an acceptably low loss.

Inherently, this creates a partial ordering among the agents as the agents which are searched first will have fewer constraints limiting their potential paths than those agents which search after other agents have already decided on a path. While this makes the solution non-exponential in the search space, the rigidity of the ordering does not allow for maximizing a throughput metric with respect to the relative ordering of the agents. Additionally, the prioritization is not explicitly defined. The proposed research focuses on this aspect of the problem: exploring methods for applying a dynamic priority based on the task, the agent, and the environment.

3 RESPONSIBILITIES

In general, it is expected that every member of the team should be involved in every aspect of the project. However, each member will have a primary responsibility in terms of the code and the report. This is an effort to ensure that all parts of the project are

	Andrew (%)	Kavinda (%)	Khushal (%)
Organization	80	10	10
Technical	35	35	30
Coding	0	0	0
Writing	40	40	20

addressed, but each member still has an understanding of the other components.

Kavinda is the main person to develop the simulator, Khushal will be focusing on the fairness and prioritization formulation, and Andrew will primarily be developing the networks and search framework as outlined in [10].

REFERENCES

[1] [n.d.]. Bootleg Fire. https://inciweb.nwcg.gov/incident/7609/. Accessed: 2021-10-25.

[2] [n.d.]. Infographic: Wildfires and Climate Change. https://www.ucsusa.org/resources/infographic-wildfires-and-climate-change. Accessed: 2021-10-25.

[3] [n.d.]. Wildfires and Acres. https://www.nifc.gov/fire-information/statistics/wildfires. Accessed: 2021-10-25.

[4] [n.d.]. Wildfires in the United States 101: Context and Consequences. https://www.rff.org/publications/explainers/wildfires-in-the-united-states-101-context-and-consequences/. Accessed: 2021-10-25.

[5] Elena Ausonio, Patrizia Bagnerini, and Marco Ghio. 2021. Drone Swarms in Fire Suppression Activities: A Conceptual Framework. *Drones* 5, 1 (2021), 17.

[6] Vishnu R Desaraju and Jonathan P How. 2011. Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 4956–4961.

[7] Connor Kurtz and Houssam Abbas. 2020. FairFly: A Fair Motion Planner for Fleets of Autonomous UAVs in Urban Airspace. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1–6.

[8] Bernhard Nebel. 2020. On the Computational Complexity of Multi-Agent Pathfinding on Directed Graphs. In *ICAPS*.

[9] Zhongqiang Ren, Sivakumar Rathinam, and Howie Choset. 2021. Multi-objective Conflict-based Search for Multi-agent Path Finding. *arXiv preprint arXiv:2101.03805* (2021).

[10] Benjamin Riviere, Wolfgang Hoenig, Matthew Anderson, and Soon-Jo Chung. 2021. Neural Tree Expansion for Multi-Robot Planning in Non-Cooperative Environments. *arXiv preprint arXiv:2104.09705* (2021).

[11] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), 40–66.

[12] David Silver. 2005. Cooperative Pathfinding. *Aiide* 1 (2005), 117–122.

[13] Jur P Van Den Berg and Mark H Overmars. 2005. Prioritized motion planning for multiple robots. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 430–435.

[14] Glenn Wagner and Howie Choset. 2015. Subdimensional expansion for multirobot path planning. *Artificial Intelligence* 219 (2015), 1–24. https://doi.org/10.1016/j.artint.2014.11.001

[15] Koping Wang and Adi Botea. 2011. MAPP: a Scalable Multi-Agent Path Planning Algorithm with Tractability and Completeness Guarantees. *J. Artif. Intell. Res.* 42 (2011), 55–90.

[16] Wenying Wu, Subhrajit Bhattacharya, and Amanda Prorok. 2020. Multi-robot path deconfliction through prioritization by path prospects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9809–9815.