HAW HAMBURG

Multi-Agent Reinforcement Learning for Adaptive Wildfire Suppression Strategies

MARS GROUP
Multi-Agent Research & Simulation

Leonard Bardtke
leonard.bardtke@haw-hamburg.de

COLOSSUS
COLLABORATIVE SYSTEM OF SYSTEMS

DLR Deutsches Zentrum für Luft- und Raumfahrt

Motivation

The increasing threat of wildfires makes the development of efficient firefighting strategies more critical than ever. In this project, we explore the use of multi-agent deep reinforcement learning to explore more effective wildfire suppression tactics. Additionally, the project analyses the relevance of communication between agents and its impact on the development of cooperative firefighting solutions.

Simulation

Given is a multi-agent-based simulation that uses elevation data from the NASA SRTM dataset and vegetation types from the OpenStreetMaps API to generate Combustibility indices. The OpenStreetMaps API is also utilized to identify the location and the size of water sources. The Meteostat API is used to retrieve historical or forecast weather information and the direction of the wind can be set at the start and during the simulation. Two types of aircraft can currently be selected: Firefighting aircraft and UAVs.
Five different firefighting strategies are currently being developed, including direct and indirect strategies.

Figure 1. Screenshot of the simulation (indirect strategy) retrieved from [1]

Methods

In order to learn the effective extinguishing of forest fires, two MADRL algorithms will be implemented to compare them with each other. For this purpose, Multi-Agent Proximal Policy Optimization (MAPPO) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) were chosen. Both algorithms follow Centralized Training with Decentralized Execution (CTDE)

MADRL-Algorithm

MAPPO (Multi-Agent Proximal Policy Optimization) [2]

- Extension of PPO for multi-agent systems
- On-policy method
- Utilizes a centralized critic with decentralized actors
- Stochastic policy to ensure exploration
- The clipping method ensures that the strategy does not deviate too much from the previous strategy during learning

MADDPG (Multi-Agent Deep Deterministic Policy Gradient) [3]

- Extends DDPG to multi-agent settings
- Off-policy method
- Each agent maintains its own policy, with a centralized critic that considers the actions of all agents during training
- MADDPG achieves the exploration by adding noise to the deterministic policy

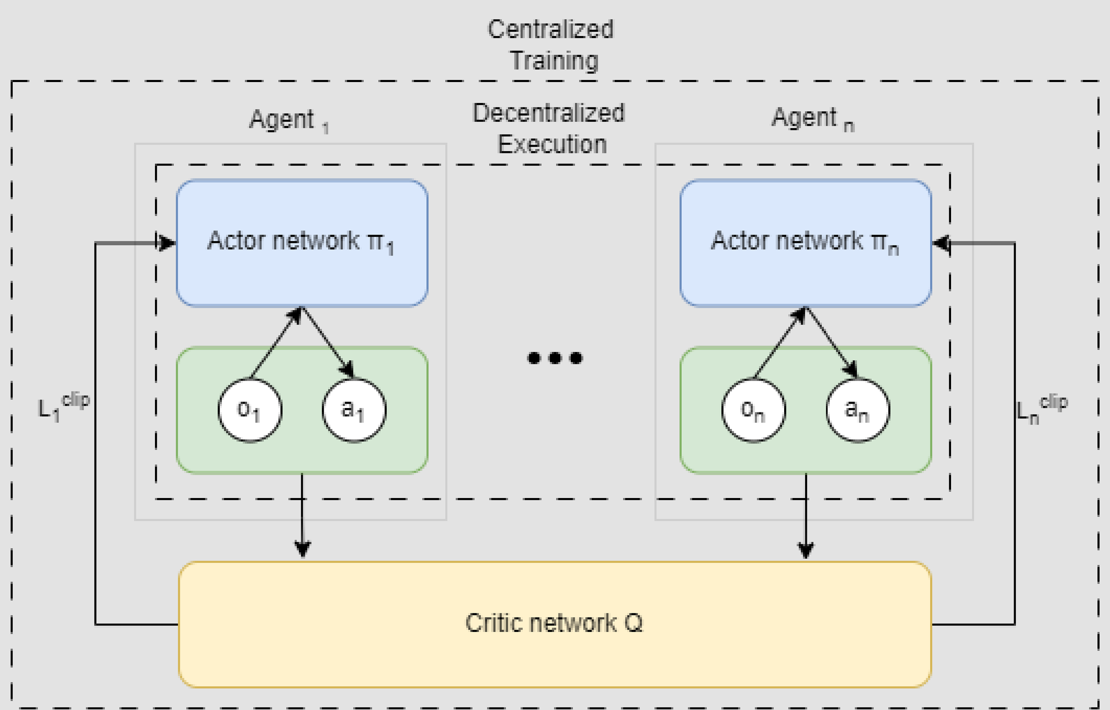

Figure 2. The architecture of the MAPPO Algorithm

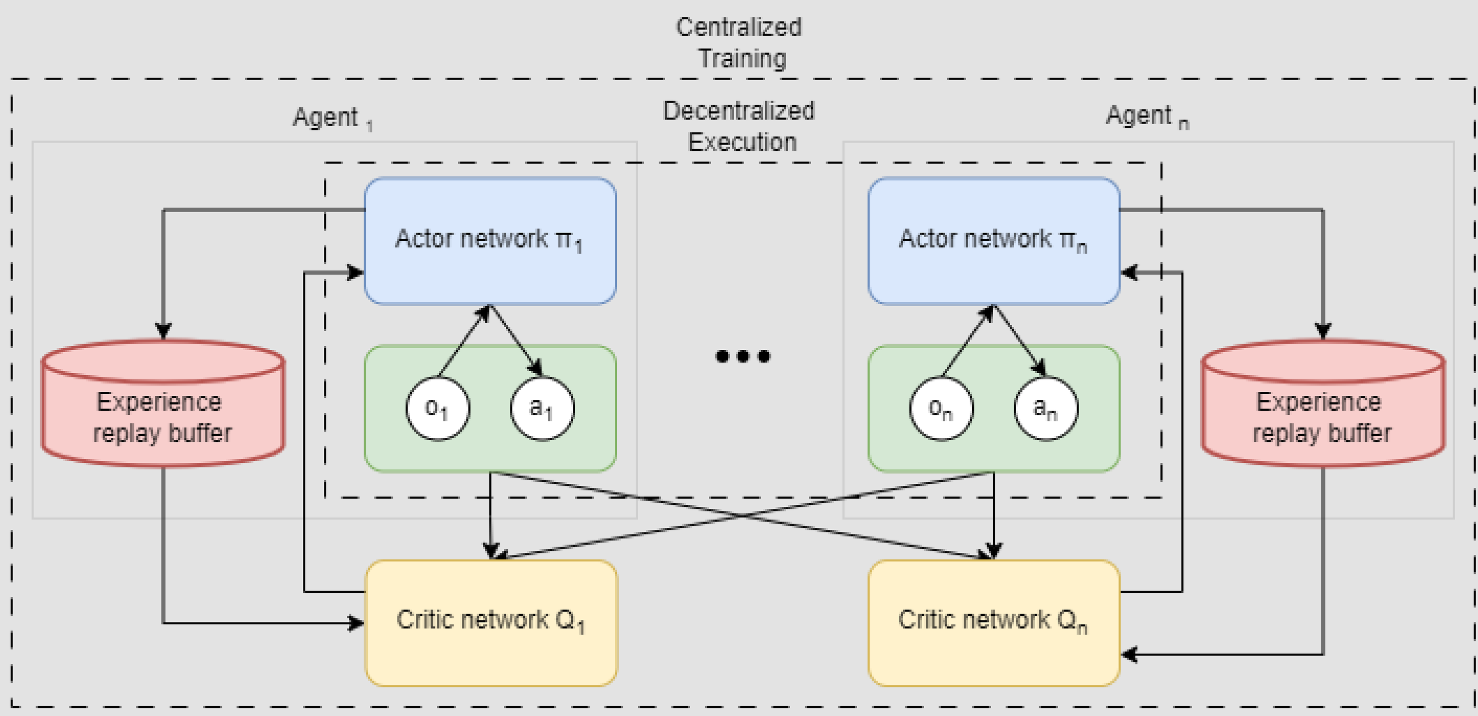

Figure 3. The architecture of the MADDPG Algorithm

Communication

Problem settings			Communication processes				Training processes	
Controlled goal	**Communication Constraints**	**Communicatee Type**	**Communication Policy**	**Communicated Messages**	**Message Combination**	**Inner Integration**	**Learning Methods**	**Training Schemes**
Cooperative	Unconstrained Communication	Nearby Agents	Full Communication	Existing Knowledge	Equally Valued	Policy-level	Differentiable	Centralized Learning
Competitive	Constrained Communication	Other (Learning) Agents	Predefined (Partial) Structure	Imagined Future Knowledge	Unequally Valued	Value-level	Supervised	Decentralized Learning
Mixed		Proxy	Individual Control			Policy-level & Value-level	Reinforced	CTDE with Individual (Policy) Parameter
			Global Control				Regularized	CTDE with Parameter Sharing
								Concurrent CTDE

To analyze the relevance of communication between multiple agents in a cooperative environment, this project develops communication methods based on the 9 aspects of MADRL with communication from [4], which can be divided into problems, communication processes and training processes

Reward

1. Evolution of the wildfire
   - How effective is the current wildfire fighting strategy?
2. Effectively usage of the water
   - How effectively is the water being used?
3. Reward the long-term strategies for wildfire spreading
   - Reward actions and strategies that will be successful in the future

Next Steps

1. Integrate MAPPO and MADDPG into the simulation and compare the resulting strategies.
2. Develop communication methods and evaluate their impact on the agent coordination.

References

[1] Kılkış, et al., (2021), "A Python Modelling and Simulation Toolkit for Rapid Development of System of Systems Inverse Design (SoSID) Case Studies ", doi://10.2514/6.2021-3000
[2] Lohse, et al., (2021), "Implementing an Online Scheduling Approach for Production with Multi Agent Proximal Policy Optimization (MAPPO)", doi://10.1007/978-3-030-85914-5_62
[3] Lowe, et al., (2017), "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments", doi://10.5555/3295222.3295385
[4] Zhu, et al., (2024), "A survey of multi agent deep reinforcement learning with communication", doi://10.1007/s10458-023-09633-6