CSE 410: Final Project Proposal

Peter M VanNostrand

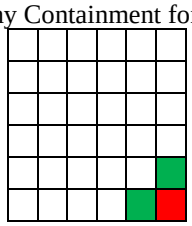
1 Topic

For the final project I will be exploring multi agent reinforcement learning in a self-designed multiagent environment. To aid with the development of this project, I will use the grid world environment that I designed in assignment one as a starting point.

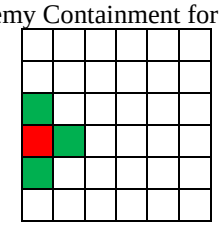
2 Objective

The objective of this environment will be for two or more agents to cooperate to contain an "enemy" by positioning themselves in the spaces adjacent to the enemy. The enemy will be a static location on the grid world that the cooperative agents must move towards to contain. For example, the following figures demonstrate possible containment schemes for n agent environments $n \in 2,3,4$ with the cooperative agents shown in green and the static agent shown in red

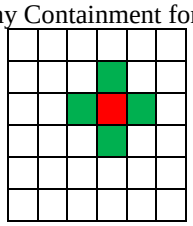
Enemy Containment for $n = 2$



Enemy Containment for $n = 3$



Enemy Containment for $n = 4$



This model could represent any number of possible real-world scenario, such as the placement of road blocks in a city to prevent the escape of a criminal, or the establishment of screening stations to contain the outbreak of a virus. The grid world will be at least 6x6 in size for a total of 36 possible states.

3 Related Work

As this project will be expanded from my grid world of assignment one, the environment will be largely compliant with the OpenAI gym standards. Instead of the environment taking a single action, it will take a vector of n actions and compute the updated environment in response to the actions of each agent.

In *Actor-Critic Algorithms for Constrained Multi-agent Reinforcement Learning Diddigi* et al. use an actor critical algorithm to train two agents in a 4x4 grid world to reach a target via different paths. This environment is similar in structure to the one proposed above, but deals with a smaller grid and a target that must be reached, rather than contained.

In *Autonomous Vehicle Fleet Coordination With Deep Reinforcement Learning* Cane Punma uses a modified Deep Q-Learning algorithm to help an autonomous ride sharing service pickup fares and deliver them to their destinations. In this case the author models the driving environment as a city grid of size 7x7 with 2 agents or 10x10 with 4 agents. The agents are then trained to collaborate to maximize their total reward.

4 Technical Outline

As stated above the environment will consist of a grid of at least size 6x6, with 2-4 agents depending on the configuration. Each agent will have five possible actions available to it, these are: moving up, moving down, moving left, moving right, and not moving. The environment will positively reward the agents for moving closer to the "enemy", and negatively rewarded for moving away from it. Once the agent reaches a square adjacent to the enemy, it will be positively rewarded for each timestep it spends there. Agents will also be prevented from exiting the grid, occupying the same tile as another agent, and occupying the same tile as the enemy. Actions that would result in all such states will be given a reward of zero.

To start a set of two agents will be used, with the enemy placed in the corner of the gird world. These agents will then be trained using Q-Learning to contain the enemy. This will serve as the checkpoint for this project as recommended by the assignment instructions. Next, I will upgrade the system to use Deep Q-Learning, this should hopefully improve the performance of our agents. Once Deep Q-Learning has successfully been implemented the task will be expanded to the three agent scenario, with the enemy placed on an edge, and finally to the four agent scenario with the agent placed arbitrarily in the center of the grid.