Artificial Intelligence

Multi Agent System

Project: UAV Fire Extinguisher

Cecilia Aponte - ID 1822225

1. Introduction

An important topic in Artificial Intelligence is the cooperation of agents to perform tasks and achieve a common goal. This project uses a simulation to manage and control wildfires using UAVs (Unmanned Aerial Vehicles) that traverse in the world with a special foam to extinguish the fires before it depletes the entire forest. The simulation and code are developed in Java using the MASON Toolkit. The task allocation protocol for fire depletion give communication constraints is Contract Net Protocol (CNP) and a Random Walk strategy for exploration task.

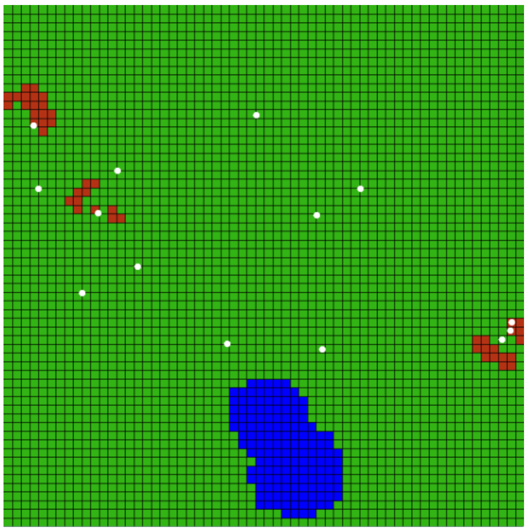


Figure 1: World Map of Simulation

2. Project Background

The forest world consists of a 2D grid, as seen in Figure 1, that has cells that can have the following characteristics: forest/normal (green), foamed and trees are safe (returns to green), burnt and unrecoverable (gray), on-fire and possibly recoverable (red), lake/water (blue). The UAV drones are characterized as one white cell and can only be aware of the current cell it is over. Due to payload constraints, the UAVs can only communicate at a limited range.

Each fire that is started, is considered as a task with its centroid where it started, the radius describing its extension. UAVs only have this two information and no knowledge of the cells around the centroid. With the use of CNP, agents select a task from the fires available. The task Manager announces the task to all agents, which then reply with an offer determined by their position and the utility of the task. The manager then awards the agents with the highest bids that are still available with the task. The agents then are all assigned to their tasks and their target as they move towards it and start executing the Random Walk (RW) to explore the neighboring cells and stop any cells that are burning. The goal is reached when either all possible cells are extinguished or burned. Since there are multiple managers, groups, and tasks this CNP task is decentralized.

3. Algorithm Implementation:

The first step for the implementation of this project is to pick the managers for each task (fire). This is picked by determining the closest agent to the fire centroid. The manager agent is then set with the corresponding task and centroid target. It then sends to all UAVs the information about its task to request for a proposal.

3.1. Sending and Receiving Algorithm

In order to send and receive messages, the agents have to check first their proximity to other agents. If it is within the communication range, the package with all the necessary information is sent. Since this posses a big issue with the development of the strategy efficiency, the algorithm was done with detail and robustness. Even though only a portion of the agents are reachable to one single agent, when put in conjunction cooperating they can form a web of link between each other where the data is transferred from the closest to the farthest possible agent given their own constraints. As seen in Figure 2, the first round of packages are sent only to those agents that are close to the 'sending agent'. Once a set of agents receive the package, these in turn use their own location to reach a farther distance within the communication range. So the task will move as follows: 'sending agent' → 'middle agent' (can retain or not the information depending on the property of the message) → 'receiving agent(s)'. When the manager sends the package, it traverses through all the levels of the agents until it reaches all agent. Each agent receiving the request for proposal from the manager stores them until it receives all task requests. When the agent is ready, it calculates its offer and sends it back to the task manager. In this case, each agent sends the packet and uses the other agents as middle agents just to transport the message until it arrives to the task manager.

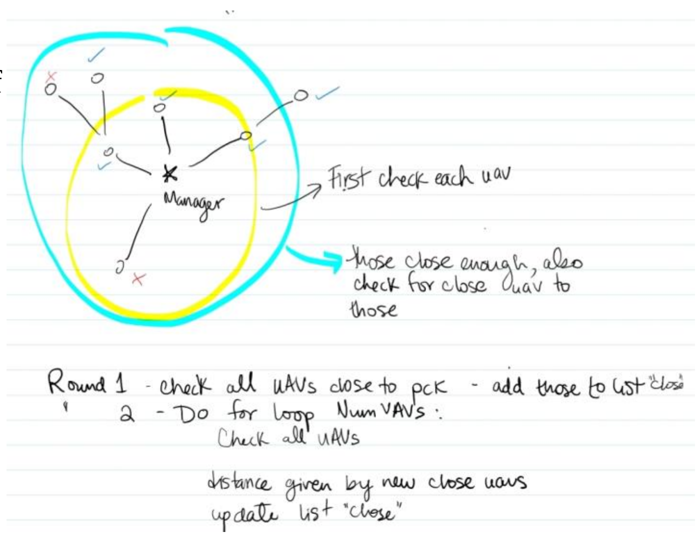

Manager
First check each uav
those close enough, also check for close uav to those
Round 1 - check all UAVs close to pck - add those to list "close"
2 - Do for loop Num UAVs:
Check all UAVs
distance given by new close uavs
update list "close"

Figure 2: Communication Strategy

3.2. Task Assignment

The agent then calculates its offer for each task using the following formula:

$$offer = \frac{total\ cells\ on\ fire\ for\ task}{\text{distance from agent to task}}$$

This formula considers two important elements that determine how well and how fast each agent can help complete the goal. The utility of the task is given by the total amount of cells under fire, so more cells on fire then the more importance it will be given by every agent. On the other hand, the greater the distance between the agent and the task will create a issue with its completion compared to closer agents. Each agent sends the proposed bids to their corresponding task manager, which then calculates how many UAVs are necessary to accomplish the task by the following formula:

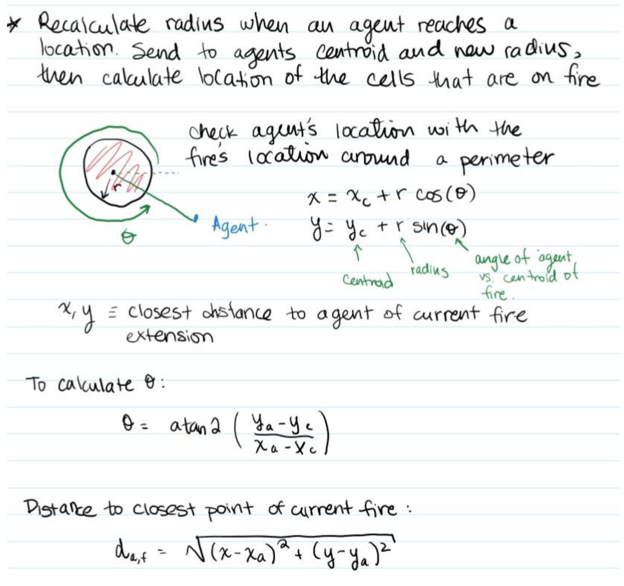

* Recalculate radius when an agent reaches a location. Send to agents centroid and new radius, then calculate location of the cells that are on fire
check agent's location with the fire's location around a perimeter
Agent
θ
x = x_c + r cos(θ)
y = y_c + r sin(θ)
centroid
radius
angle of agent vs. centroid of fire
x, y = closest distance to agent of current fire extension
To calculate θ:
θ = atan2((y_a - y_c)/(x_a - x_c))
Distance to closest point of current fire:
d_a,f = √((x - x_a)² + (y - y_a)²)

Figure 3: Target calculation for agents

$$\#\text{ UAVs needed} = \frac{\text{total cells on fire for task} * \#\text{ UAVs}}{\#\text{ fires}}$$

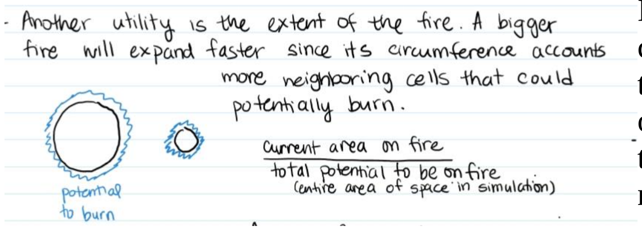

Another utility is the extent of the fire. A bigger fire will expand faster since its circumference accounts more neighboring cells that could potentially burn.
potential to burn
current area on fire
total potential to be on fire
(entire area of space in simulation)

Figure 4: Depiction of rapid increase of exposed cells as radius increases

In this case, more UAVs available will correspond to more needed, just as the total cells on fire. Since there is more than one fire, it has to be divided between these. The highest bidders that still have not been assigned to a task are given those tasks. The target of the agents, however, are different than the managers who went directly to the centroid. Since the center of the start of the fire is taken care by the manager, the agents have to be able to contain the fire also from the outside. As seen in Figure 3, it is very important to be able to contain the fire from the outside since as the radius grows so does the perimeter that is exposed to being on fire and from growing exponentially to a point where it is impossible to recover given the amplitude area. To calculate the target location as seen in Figure 4, the manager calculates the location at the perimeter of the fire radius that is closest to the agent's location. This way the agent does not have to maneuver far, but rather a radius-away from the centroid in the direction (angle) of the agent with respect to the fire centroid. A small amount is added to the radius to account for the expanding radius/area until the agent arrives. It is incrementally added, so the last agents are given a bigger radius addition. In this way also, the fire is attacked in different levels not only the center or the outside but also the middle portions as seen in Figure 5.

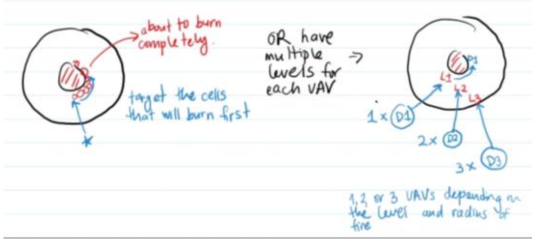

about to burn completely.
target the cells that will burn first
OR have multiple levels for each VAV
L1
L2
L3
D1
1× D1
2× D2
3× D3
1, 2 or 3 UAVs depending on the level and radius of fire

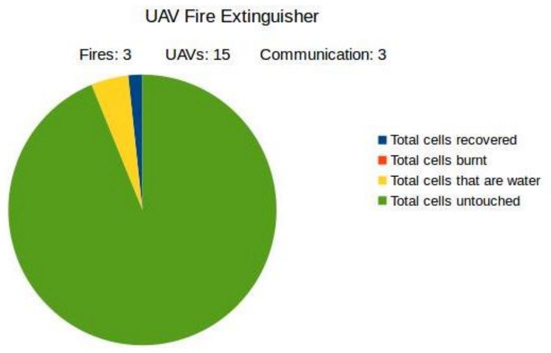
Figure 5: Levels of target locations to extinguish fire efficiently

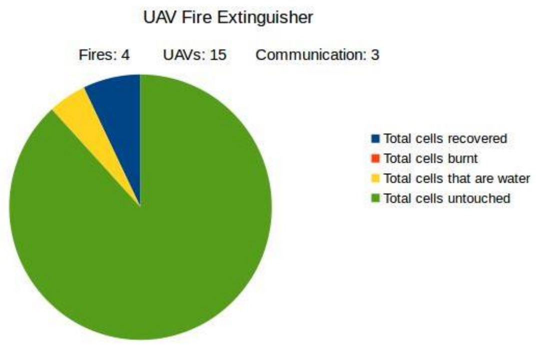
3.3. Random Walks

The random walk is an approach to explore nearby cells in order to determine they are needed to be extinguished. Initially, the algorithm visits only the cells that are next to the current cell (1 cell in any direction). If the cells are not on Fire, more attempts are done but now the exploration is farther away. Two implementation strategies are used. The first was a constant increment of 2, which had very successful results. However, the algorithm would sometimes find itself stuck in a loop and was not able to terminate. When this was corrected and the random range increased as the attempts did also (directly correlated) and the loop was closed, the results gave a worse outcome.

4. Evaluation Results

As explained previously, two strategies were implemented in the Random Walks. The first one was most successful yet not robust with repetition. On the other hand, the second one was less successful but robust. In the next section, improvements will be discussed. Different experiments were performed with both strategies. The initial and most successful trial was using 15 UAVs, 3 fires, and 3 as communication distance. The results, shown in the top left of Figure 6, show perfect containment with NO burned cells in all 50 trials. This is compared to the same strategy but with 4 fires, showing more recovered cells. The bottom graph on the other hand shows the result of the 2nd strategy. Most cells show as burned and only a few as recovered.





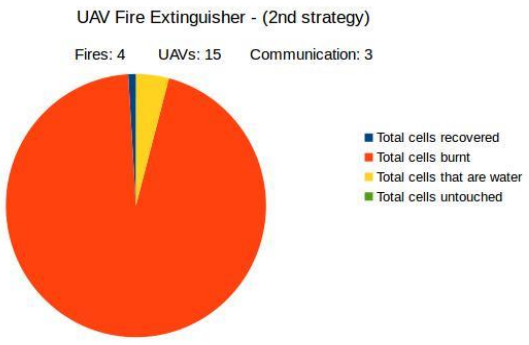


Figure 6: Top Left - 3 fires Top Right - 4 fires; Bottom: 4 fires with 2nd strategy

Another comparison made is the change due to communication range. Given the more efficient strategy, an increase in communication range does not change much. On the other hand, the number of UAVs does make a significant difference when moving from 9 to 12 UAVs where in the former most cells are burnt and, in the latter, almost all are untouched. However, not much difference is seen when increasing from 12 to 15 or 20.

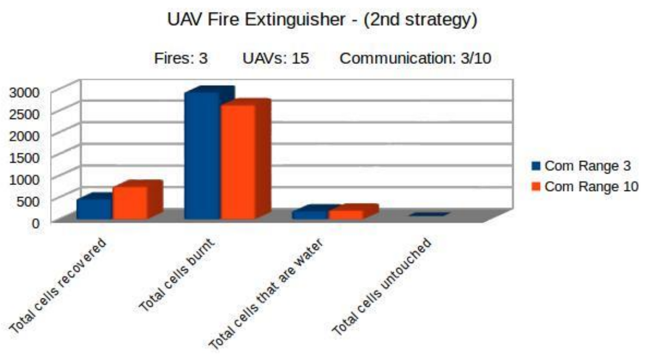


Figure 7: Minor changes seen with changes in communication range from 3 to 10

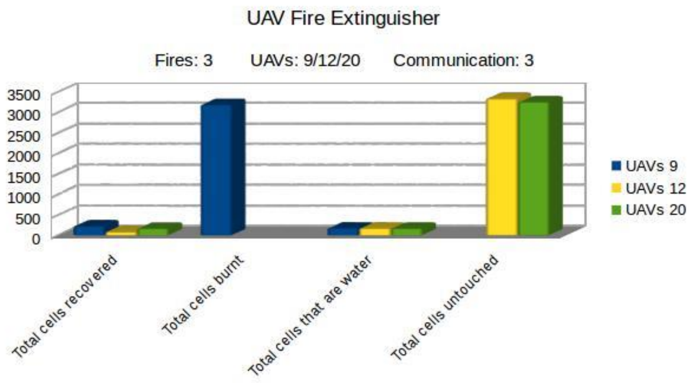


Figure 8: Changes seen when increasing UAVs

5. Conclusion and Further Improvements

Given the results from the experiments, it is evident that given the current algorithm the best parametrization is 12 UAV, 3 or 4 fires, and 3 communication range. Improvements would be valuable by correcting minor bugs and adding other options such as adding proximity of water to determine UAV needed, updating the task utilities more often, and using the data of already seen cells to move in the random walk and potentially share with its task manager.