# Coursework 2: Forest Fires and Cooperative Agents

**Task 1**

**Part (a) (max 500 words)**

<u>Messages that get exchanged and description of the protocol:</u>

The first message that is sent is created by the sensors, it is sent to all the fire units and contains the location of all the fire (“inform-all”). The message is sent when the scouter reaches an active fire. This is then used to work out the closest fire location, which is used when the closest fire-unit has been found

The next message that gets exchanged between the agents is sent from a fire unit to all the other sensors and all the units (“send-all-unit-locations”), the message is sent when there is a belief of a fire for the units. The message contains the location of all the units this belief about the location is created within a list in the message, it also contains the sender of the co-ordinate, this is so the sensors know which unit has sent the coordinate.

The final message that is exchanged is sent from a sensor to all fire agents (“send-closest-location”), the message is created when the sensor detects a fire and knows that there is fire agent close. The belief about which agent is closest is determined in the ‘closest-agent-belief-class’, this class finds the agent that is closest to the sensor by getting all the beliefs from the unit-location list cycles through the list comparing the closest agent in the list and the new agent it is comparing it against by calculating the manhattan distance between the sensor and each agent. After it has compared all the agents it will report back the location of the closest agent.

<u>Why you believe your proposed protocol will be effective.</u>

I think this protocol will be effective as when the scouters know about a fire the fire-units will be able to work out which one is closest and send it towards the fire location. This is an effective technique as it will minimise the number of fire-units heading in the same direction, allowing for a wider spread of agents across the map and in turn increasing effectiveness of model. This protocol is also increasing effectiveness as there is a check in the intentions that means that a job cannot be allocated to a fire-unit if it already moving towards a fire, this means that if the unit is on a mission already the sensor will look to find a new one instead of waiting for the closest unit to become available leaving less time for the fire to be burning before an agent is at the fire and ready to extinguish it.

**Part (b) (max 400 words)**

<u>A description of your implementation of your proposed communication</u>

To implement my protocol the main objective was to work out how to find the closest unit to the fire, this was achieved in a similar manner to the way the closest fire location is worked out for the units. When I found out this location it was important to use it in the right messages so that both the sensors and the units knew which fire agent was closest to it. When the fire-units had the beliefs of the closest unit it could then start the process of moving the unit towards the fire given it knew the fire location and it was free to do so *(““if exist-beliefs-of-type "Closest Unit" and exist-beliefs-of-type "fire" and current-intention = "not-be-moving"”*”). I worked out the unit that corresponded to the location given earlier by comparing each fire-units location to said location ("*if current-location = closest-belief*"). From here it would search for the fire and then move towards it. To make the protocol more effective I have enabled patrolling for the fire-units as this allows them to reactively be putting out fires when not being assigned a job.

For the scouters, the implementation was straightforward. The main change that was made to the scouter's procedure was the enabling of the ability to receive messages ("*collect-msg-update-intentions-sc*") and update intentions ("*update-intentions-sc*"), this was crucial to the protocol as these allowed scouters to send the closest location to the units, in turn setting off the movement process. One change I did make was that they will not wait at a fire until there is a fire-unit there to extinguish it, as the intentions (of a fire location to travel to) will be added into the stack I have enabled them to be moving around the map after being at a location for a brief amount of time, this will still have the same effect of adding intentions to the fire-unit but will allow them to add more should they find the fires before the fire-units

**Task 2**

<u>What experiments did you run?</u>

For the experiments some constants in the parameters were set; tree-num is 300 and number-of-fires is set to 40. To test they effectiveness of my system I would change number of scouters, fire-units and initial water. For the first experiment (Experiment A) I decreased the number of scouters and increased the number of fire units, keeping initial water 25. The next experiment (Experiment B) I increased scouter number and decreased fire units, keeping the initial water at 25. Another experiment I did was to have an average scouter number, low amount of fire units and lower initial water (Experiment C). Finally, I tested scouter number at 20, fire units at 25 and initial water at 25 (Experiment D). I will run each experiment multiple times and plot the average number of dead trees and/or saved trees for each parameter that is changed

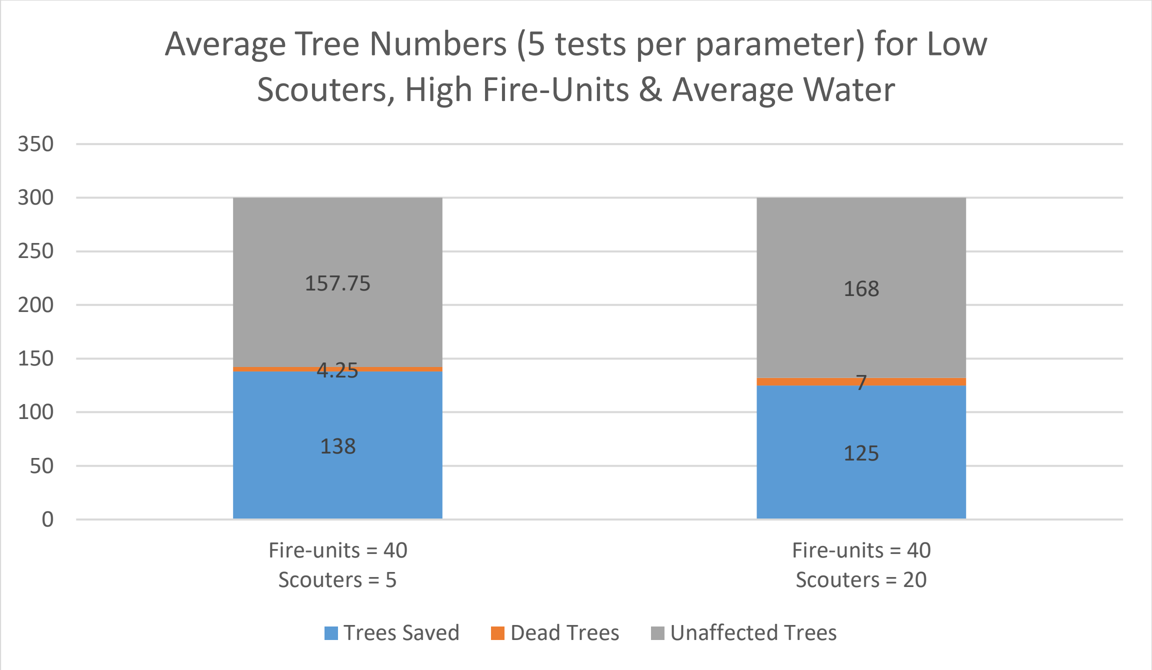
<u>Motivation for experiment design</u>

The reason I tested the parameters for the first test was to see if the scouters were the most important part of the simulation by lowering their number and increasing fire units meaning there will be as many fire units in the simulation as fires. So, comparing this simulation where they cannot move until they have been assigned a job to the naïve model, the amount of fire units may not impact the success so much. The second test was to realise the impact of fire units on the simulation, as there would be plenty of messages sent to each fire unit, this test was to see how effective a small number of fire units would be when they are assigned a close job each time, compared to the naïve model where the fire units moved at random.

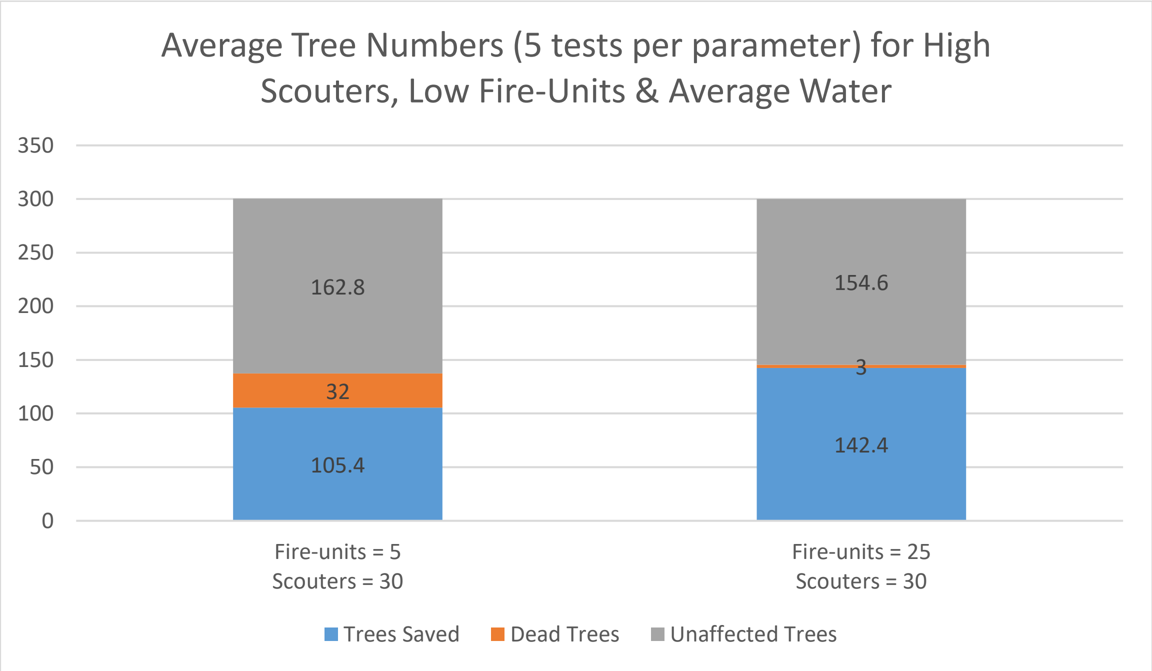
The third test was to see if the amount of water the units carried effected the success of the model when there was a wealth of scouters in the map but low amounts of fire units, having a lower amount of water in the fire units means that they would have to return to the base more often, but they would move around the map faster. Compared to the naïve model, this may be more detrimental as the fire units will be near the base more often meaning the fires that start on the edge of the map may not be reached in time. The final test was to see how the simulation went when there was a wealth of all the parameters available to the model, this will help draw a comparison of the effectiveness of the extended model against the naïve model.

Results

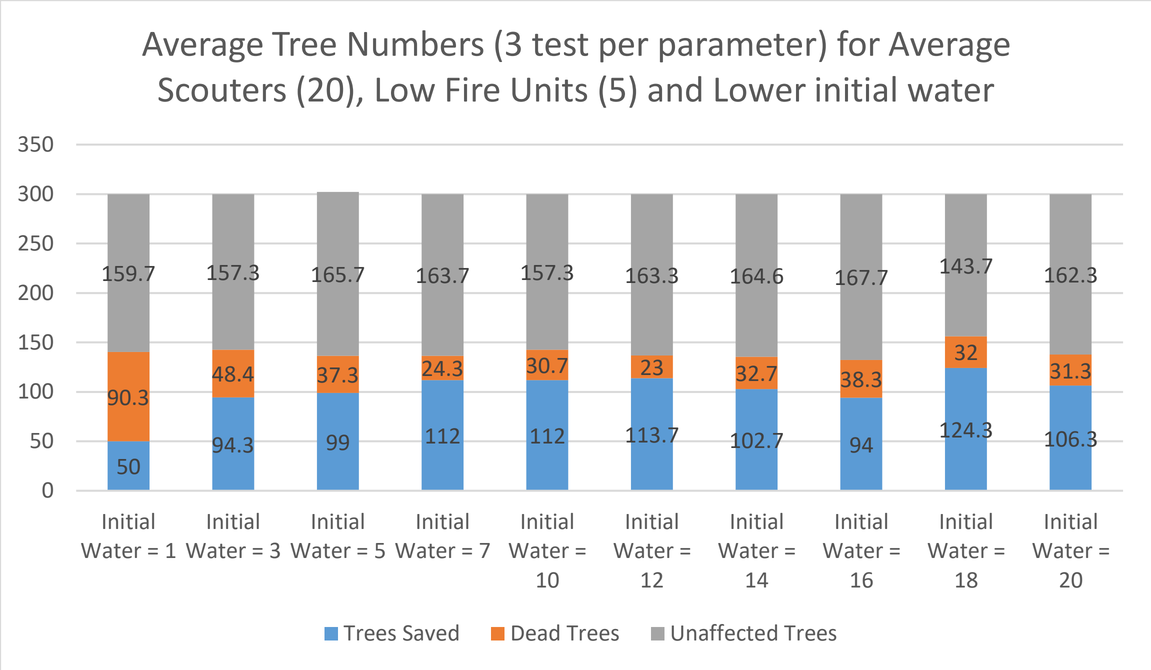
Experiment A:



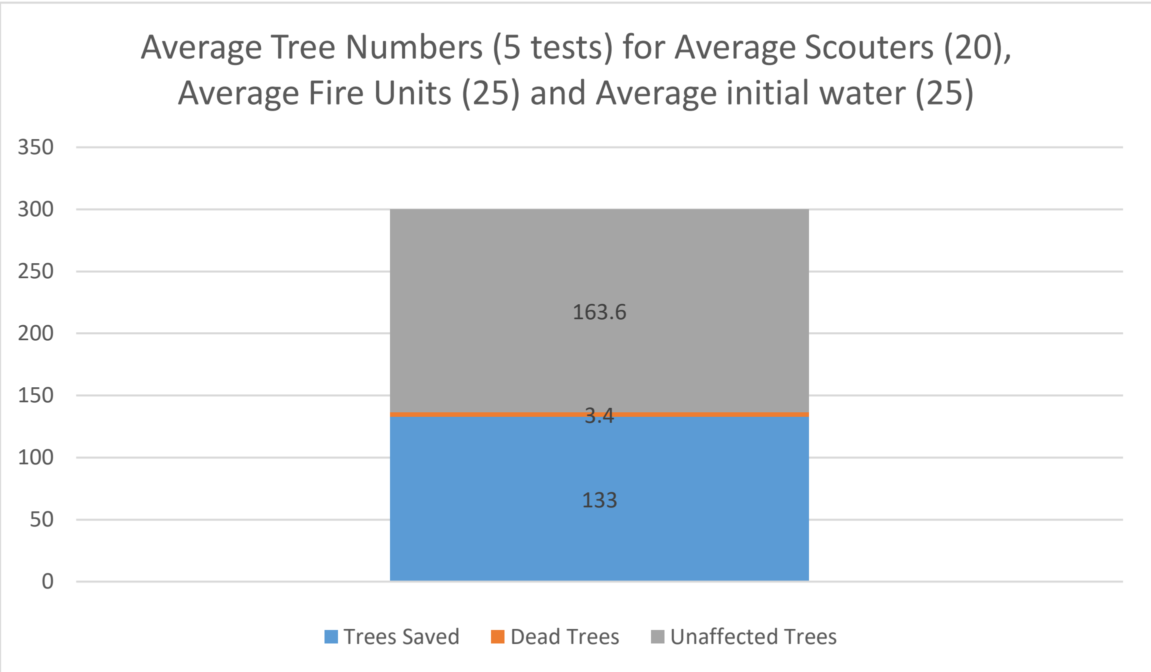

Experiment B:



Experiment C:




Experiment D



Analysis of your results

Experiment A – From these results we can see that even with low numbers of scouters, if an average number of water is available to the units then the model is still successful, this could be because my scouters will continuously add to the stack without stopping so lower numbers shouldn't affect the amount that die. We can also see that the number of unaffected trees is larger on the result with more scouters, this will be because if there are more scouters, they should be able to add fires to the stack quicker.

B – From these results we can see that when there is a low number of fire-units then the model isn't as successful, this could be because when they have an initial water of 25, they will be slow and as there is not many fire-units they will struggle to get to the fires.

C – From these results we can that overall, the model is still relatively successful throughout the test when there is a low amount of water on each fire-unit. The optimum seems to be between initial water of 7 to 12, this could be because the unit can still move quickly and not need to go back to the base to refill before all the fires have been extinguished.

D – From these results we can see that when there is a wealth of each parameter available to the model it is very successful, on some tests it came back with 0 dead trees. In comparison to the naïve model I would say that they are both successful, but the new model may take a little bit longer to extinguish all the fires

Limitations of your results

One limitation to my results is that the tree-number and number-of-fires are fixed. This means that we cannot explore the effectiveness of the model on a smaller terrain or a smaller isolated fire, these would be useful as they emulate a large percentage of fires in the real world. Another limitation to my results is that there is a large jump between the amounts for fire-units and scouter numbers. This means that we would have get an estimate for the results in between these jumps, which is unreliable. A final limitation for my results is that there is still some random movement for my fire-units and fires are allocated randomly, this means that the results can vary dramatically between tests depending where the agents 'patrol' to and depending on where the fires are started. If they are close then we will get better results, if they end up being further away we will get worse results

**Task 3 (max 300 words)**

What are the advantages of using hybrid agents in this domain?

One advantage of using a hybrid agent in this domain is that that the agent can maintain the belief of a fire location in its stack throughout the process so if it is told that there is a fire somewhere it can extinguish fires on the way to the belief or if having to reactively refill on water, this has also allowed me to keep the scouters moving throughout the model. Having a hybrid agent also allows the agent to be moving randomly across the map whilst awaiting a message from a scouter about a definite fire location, meaning that some extra fires may be extinguished before any sensor has encountered a fire, again, improving the efficiency of the model.

<u>What problems might you anticipate in this domain if the agents were modelled according to a pure BDI architecture?</u>

One problem that may occur if there was a pure BDI structure would be that the agent may receive a fire location before the agent receives a message containing the request to return to the base because it no longer has water, this could lead to the agent getting to a fire and not being able to extinguish. This, in turn, would also mean that the unit would be stuck at the fire location unit it has gone out as it wouldn't be able to remove it from the stack, decreasing efficiency. Another problem I would anticipate with a pure BDI architecture is that if an agent has a fire location in the stack, but the fire goes out, it will keep moving towards the fire location anyway without being able to divert course. This is because it has the intention of travelling to the fire location in its stack which is unaffected by if the fire is active or not, this would mean it is not able to extinguish any fires for this whole duration as there is no reactive element that allows extinguishing en route, this again decreases efficiency of the fire units.

END OF COURSEWORK SUBMISSION