|  |
| --- |
|  |

AMS Coursework 2

Fire extinguishing agent

Ali Azam | BSC Computer Science | 1528624

# Introduction

The report will discuss the strategy used for implementation, challenges faced, learning curves, experimentation and combination of multiple agents. The report will mainly cover the following parts:

1. Extended Agent
   1. Description and Justification of the design
   2. Description of implementation
2. Experiments
3. Results
4. Advantages of Hybrid Design

# Extended Agent

1.a **Description and Justification of the proposed protocol:**

The extended agent still uses the BDI architecture and extends it to add more features such as communication and further beliefs about the world. The agent design is as follows:

The role of the ground units to put out fire and be hybrid agents which act both reactively and proactively, whereas the role of the scouters is to search for fires and inform all the agents about the fire, the design of the agents have not been altered, their roles remain the same as they were, only the communication, beliefs and intentions are added to increase their efficiency.

In a nutshell, the design of the cooperation was kept simple and intuitive to make it extendible and efficient. Most of the things were kept the way they were and only minimalist change was done to increase the efficiency of the program.

The idea is very simple, the scouters should inform all the ground units about the fire and the ground units should still believe there is a fire at a certain coordinate, but not take any action until instructed to do so by the scouter. How does the scouter make the decision on whom to instruct to eliminate the fire? The answer is simple, it chooses the best one based the distance. The question becomes, how does the scouter know about the distance of all ground units to itself? That’s where FIPA communication kicks in and solves the problem.

The communication works as follows:

**Scouter:** Inform all ground units about the fire.

**Ground Units:** Calculate the distance from itself to the fire, create a reply message and add the distance as the content of the message and send the reply

**Scouter:** For every incoming message from the ground unit, check if that’s the minimum distance its seen in the content of the message, if yes then update the minimum seen and record the message, if no then ignore it as it already has closest agent who is best fit to extinguish the fire.

**Scouter:** Once iterated over all messages in the incoming queue, create a reply message and add the contents of the message which will be the fire locations and mark it as fire-locations-to-put-out and send the reply

**Ground Unit:** The specific ground unit will receive a response and it will add fire location to its beliefs which will then turn into its intention because of the label fire locations to put out. (Further explained below in 1.b)

The motivation of this solutions comes from various day to day processes such as Job application process where the company X advertises their job everyone, then applicants apply for the job from all over the world and X prioritizes applicants who are the closest since they can be the fastest to join. Also, keeping in mind the Hollywood Principle, [http://wiki.c2.com/?HollywoodPrinciple] “*Don't call us, we'll call you".*

It was hard to decide at the earlier stage of the design process on whether the ground agents should be the one to allocate themselves the job and let others pick other jobs, however this wasn’t a feasible solution as if the ground agent which picks it is surrounded by other agents then it will be stuck there and the tree will die.

1.b **Description of implementation:**

The above description of the protocol gives a brief overview of the implementation and each part of the implementation will be discussed here in detail explaining the decisions made and justifying them.

The first part of the code which was modified is adding the “move-randomly” to the ground agent as the tests showed that when the agents were moving randomly, the trees on fire were put out much faster than when “move-randomly” was not in the agent.

One of the important design decisions made here is that move-randomly was called within the units behavior and not added as an intention, the reason for this is because the agent in my opinion should not have the intention of moving randomly but move randomly when it has nothing better to do, this is more like humans, we wonder around not really looking for something but just wondering around without an intention. (In this case, the intention of “find-target-fire” will be there). With the design difference aside, the behavior of the agent will remain the same in either way.

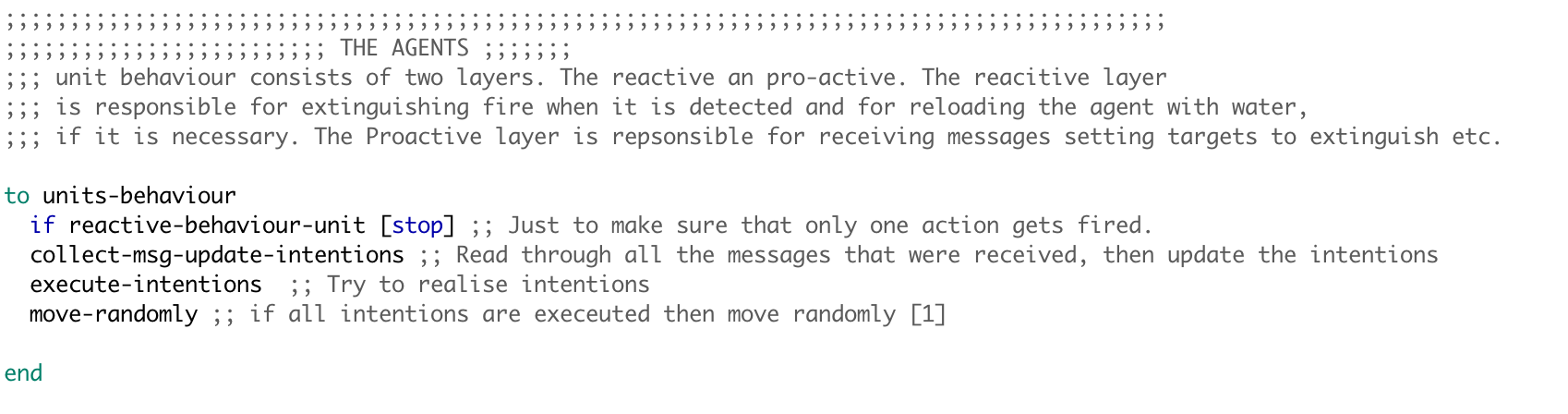


Figure Showing the addition of calling move-randomly to make ground units move randomly

The scouter behavior at the start remains the same, it moves around to look for fire and once it has found a fire, it informs all the ground units about the fire and the content of the message is from a function **fire-location-s** (this will be addressed later in the report). The ground units get the message of the fire and they add it to their beliefs as they used to, and they calculate the distance from the location in the message (content of the message) and from their coordinates, with the closest distance, the agent creates a reply message (FIPA library function which automatically adds the sender and receiver) and adds the content to be the distance and sends the message. [Figure 2]

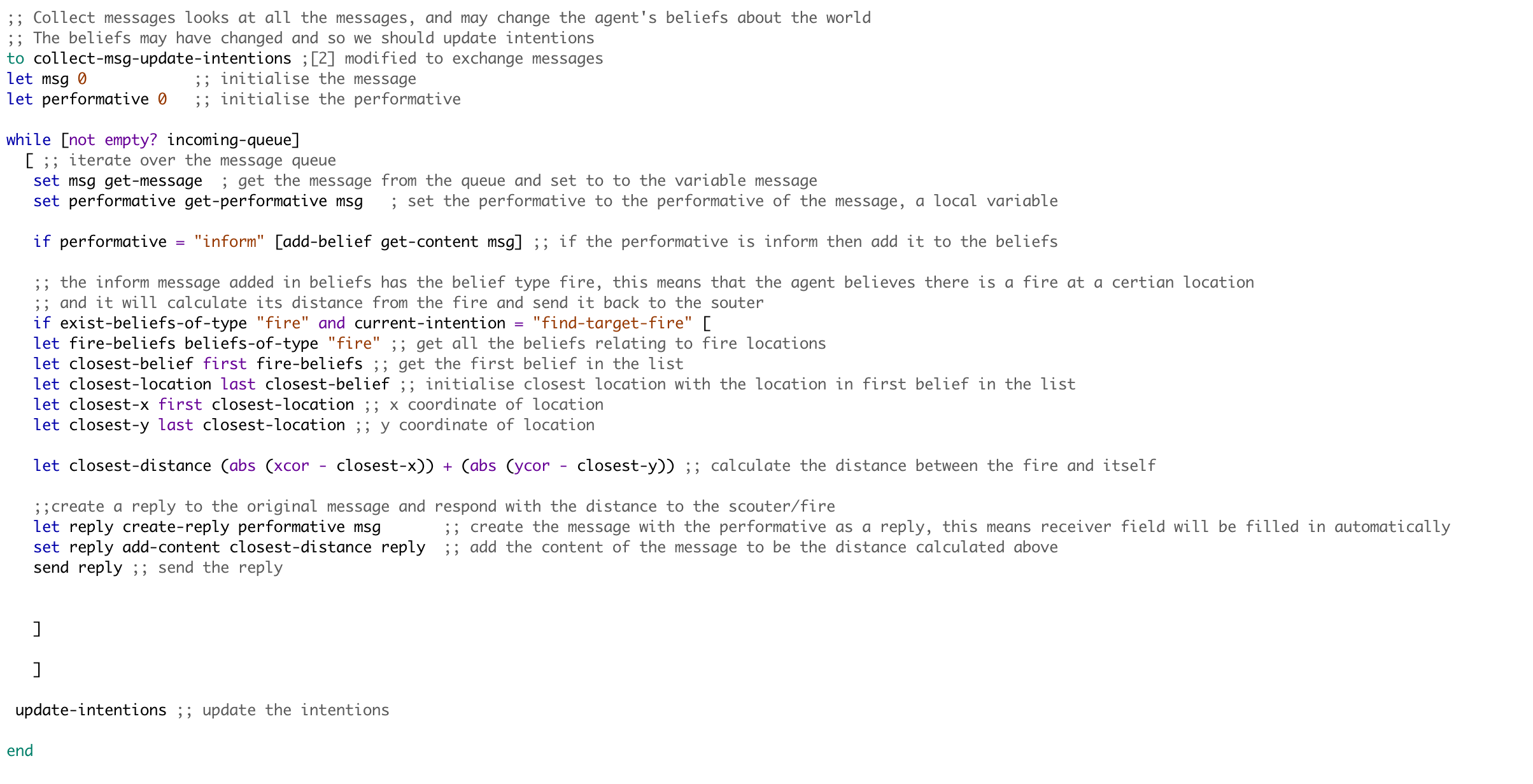


Figure Showing replying of the broadcast message with their distance

The message sent by the ground units are received by the scouters and it goes through all the messages storing the minimum distance seen and its respective message, once it has iterated over all the messages and knows the best agent to command to, it creates a reply message and the contents of the message are from a function “fire-locations [Figure 4]” and sends out the reply [Figure 3]

There are several design choices made at this point, one is that a reply is sent straight away after finding out the best agent to put out the fire and beliefs of the ground unit are used to make this possible [Figure 5 and 6] and with the help of the a newly created function [Figure 4].

The considerations were to have a similar design as ground units and messages are added to the beliefs or the best message is added to the belief and the intention would then be a function which will send out the message based on the scouters beliefs, while this would be a feasible solution, it will take away the simplicity and the argument that I had with myself is whether replying to a message should be an intention or not and the decision was in the favor of keeping it simple and straightforward. Although this would not have any implication of the performance/efficiency of the program and hence was kept this way. [Figure 3]

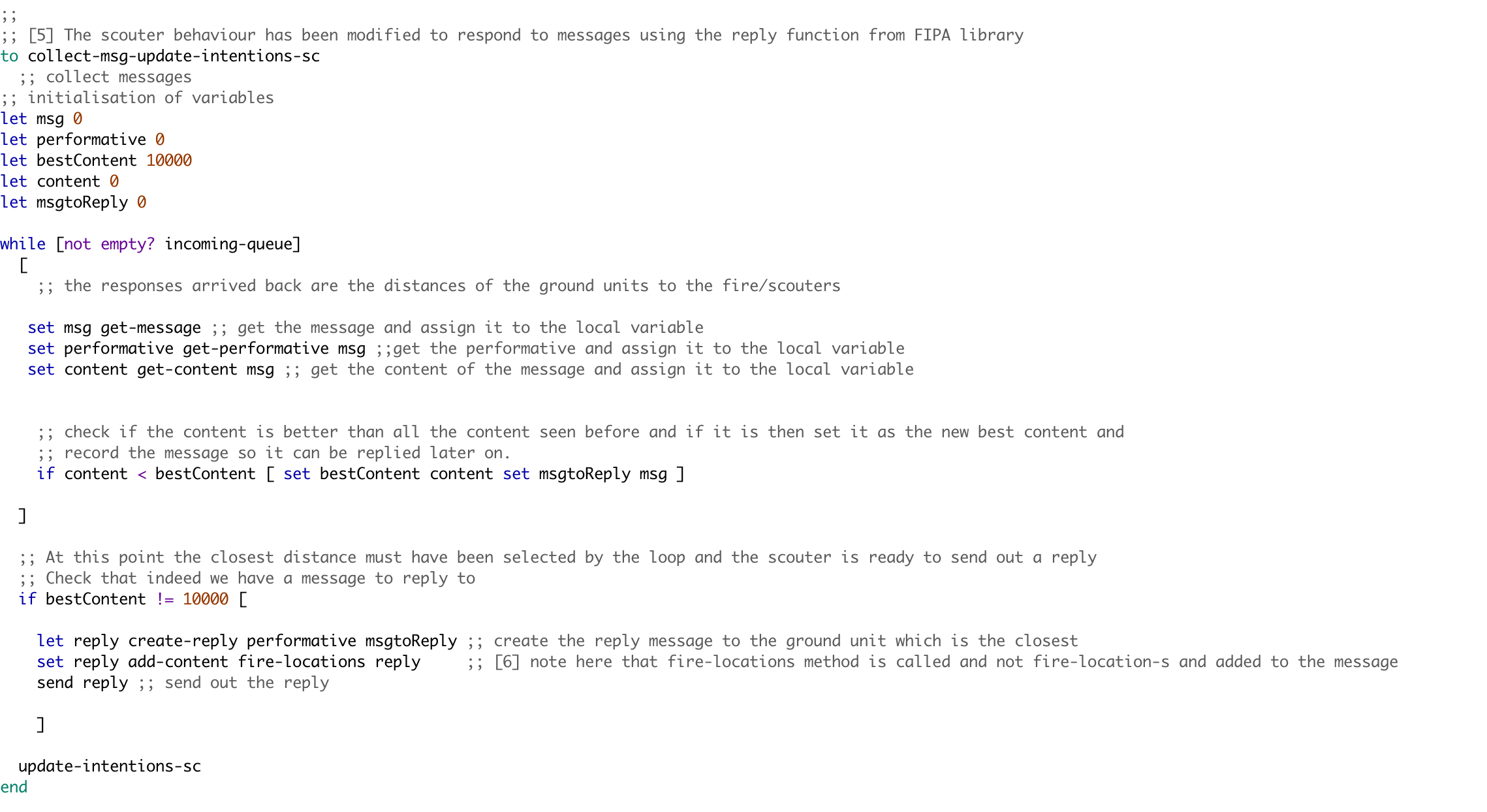


Figure Showing the altered code in collecting message and updating intentions of the scouters

The name of the list was altered in this function to distinguish between the fire coordinates requested for distance and fire coordinates assigned to the specific ground agent to extinguish the fire. [Figure 4]

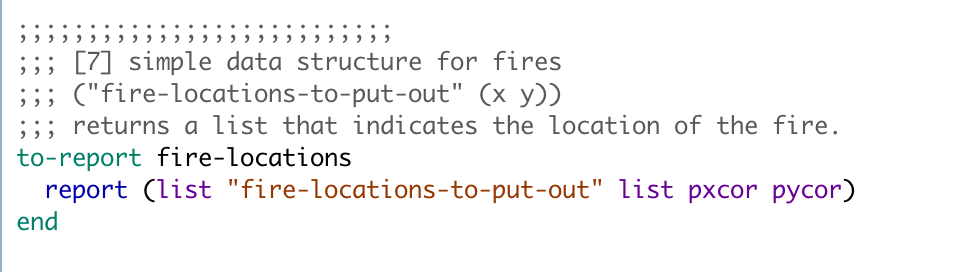


Figure Showing a new report mechanism added to aid the BDI design

Once the function was altered, this meant that the ground agent will have 2 types of beliefs about the world, one is about where fire is in the world and second is fire-locations-to-put-out which are the fire locations assigned to the agent, now this meant that update-intentions needed to be altered to make it work which is shown in [Figure 5 and 6].

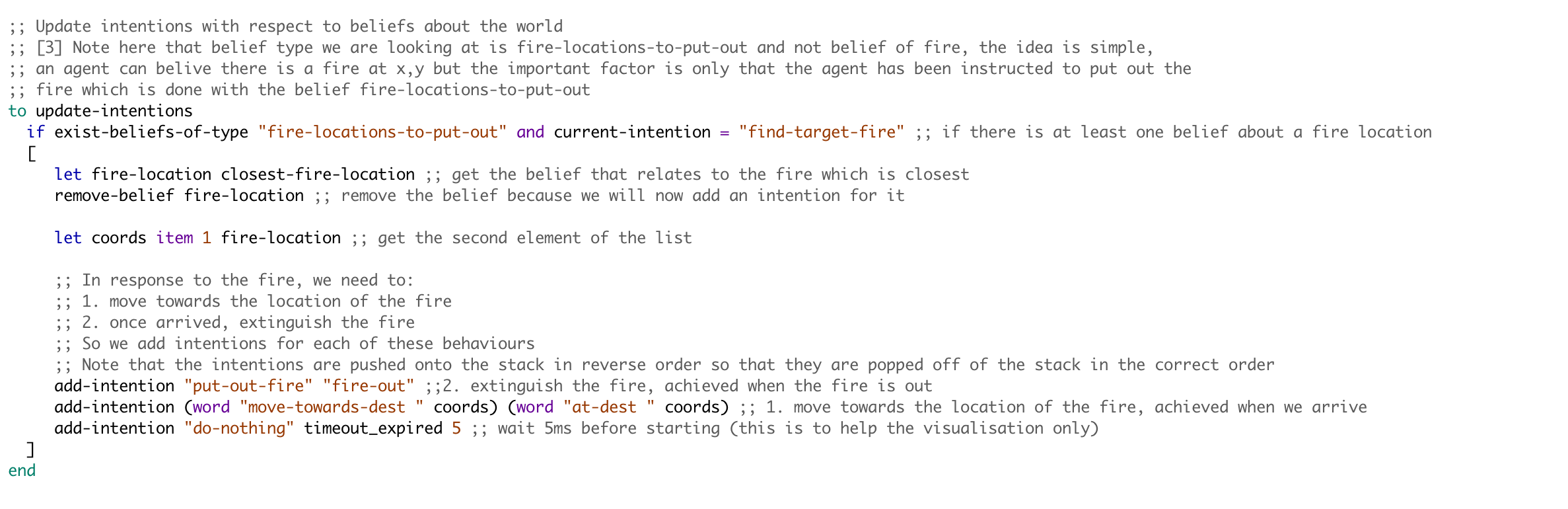


Figure Showing the alterations in the update intentions of the ground units

For the update intentions to work as expected, Closest-fire-location had to be altered as well to only get the beliefs of type “fire-location-to-put-out” and once this was done, the cycle of communication was complete and the agents were able to go in different directions as they were commanded by different scouters for different fire.

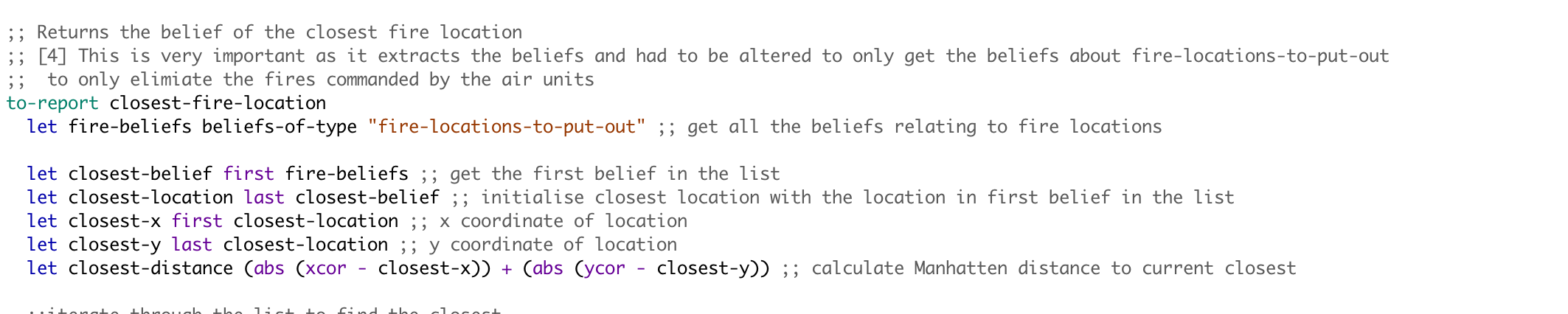


Figure Showing alterations in the closest-fire-location procedure

It was also considered on whether the scouters should start searching for next fire after reporting the fire or should stay there but the test results showed that the performance was quite similar and since I have the move-randomly for ground units, their reactiveness takes care of a lot of trees.

To wrap up the implementation process, several design choices were made throughout the design and implementation process and a lot of them don’t influence the performance or efficiency of the program and purely matter as how we perceive.

**Description of the experiments conducted:**

There were several tests conducted to evaluate the performance of the extended model and the experiments are as follows (Initial-water was set to 25 in all cases):

**Experiment: 1:** Fire-unit-nums: 1 Tree-num: 400 Number-of-fires:40

This experiment focused on testing what happens if there is only one unit to extinguish the fire as the worst case scenario and to test whether it refuels when it runs out of water and general performance in terms of how randomly moving around will distinguish fires. The reason 400 trees were selected is because the agent will have a better chance of finding the trees on fire if there are a lot of them. (10% of them were on fire).

**Experiment: 2:** Fire-unit-nums: 1 Tree-num: 100 Number-of-fires:40

This experiment was similar to the experiment 1 but number of trees were set to 100 which was the minimum it could go and the reason for this was to see how the agent performs when the environment/trees are very sparse and it was found that one agent could only save 10 trees on average with these parameters as the trees were far away from each other and also the agent did not detect the that two neighboring trees were on fire hence it should have put out both but it moved randomly instead.

**Experiment: 3:** Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:40

This experiment set the maximum of everything to see how the agents behave in terms of coordination and whether they will be able to put out all the fire together as they all had 25 units of water each and trees were quite close to one another, the result of this experiment was that dead trees were an average of 5 and in some cases even 0, hence an average was used.

**Experiment: 4:** Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:1

This experiment was used to test what happens if the fires start out with 1 as its often the case with the real world that the fire starts with 1 tree and then goes wild. This was tested with all 40 agents to see whether they can control it before it goes wild and becomes uncontrollable, this was kept in mind that 40 agents may not all be present at hence experiment 5 was carried out to support this.

**Experiment: 5:** Fire-unit-nums: 10 Tree-num: 400 Number-of-fires:1

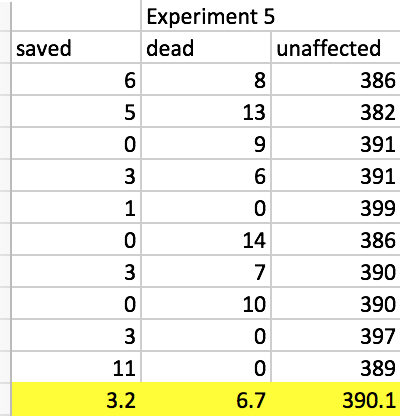
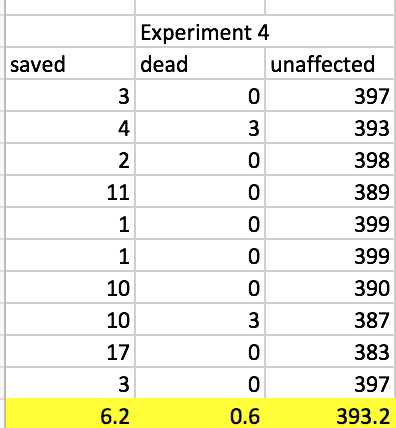
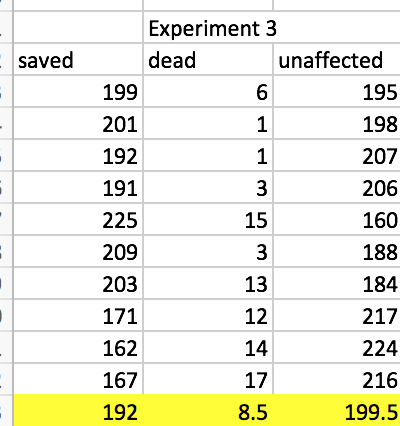
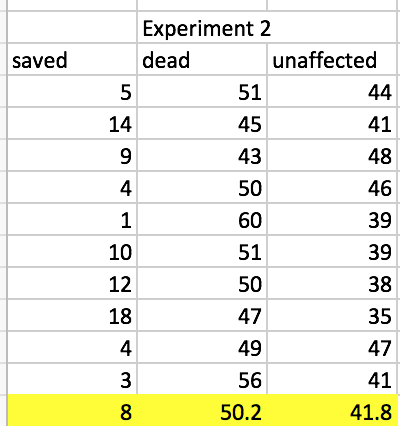
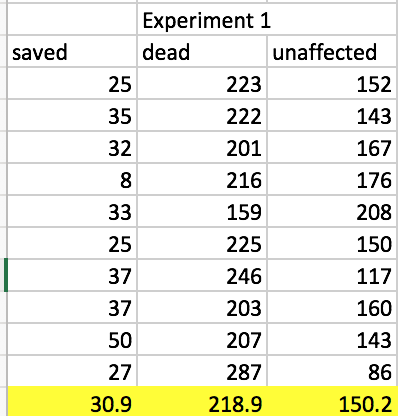
This experiment was carried out to see whether if the fire starts out with only one fire and there are 10 agents deployed in the forest for precautionary measures will be able to handle it as it happens in reality that at the start very few agents(humans) try to extinguish the fire even with limited resources and they often result is saving the forest. In this experiment the extended agent model performed well as the average number of dead trees were 10 and in most cases the agent was able to control the fire earlier on and dead trees were reduced to minimum.

**Results:**

The results are an average of running the simulation 10 times to get a better estimate rather than running it a single time. (The individual results are shown below)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Saved trees | Dead Trees | Unaffected Trees | Analyses |
| Experiment 1 | **30.9** | **218.9** | **150.2** | For one agent in the whole forest and 40 fires to begin with, it is good that it can save almost 31 trees as roughly 60% trees caught and almost 10% were saved with just one agent. Although the number of dead trees are significant, it cannot be blamed to the agent as there was only one and there were 40 fires. The conclusion drawn from this is that the agent works really well as alone it saves 10% of burning trees on an average. |
| Experiment 2 | **8** | **50.2** | **41.8** | This time around the trees were quite sparse because there were only 100 trees and 40 of them were on fire, the agent did quite well with saving 8% of trees with almost 60% on fire. This shows that perhaps 8 agents are enough for a forest of 100 trees as one can eliminate 8% of the total fire on average (based on simulation) |
| Experiment 3 | **192** | **8.5** | **199.5** | Best case scenario based on maximum values allowed in the simulation. With all 40 agents on the field, they did a pretty good job by eliminating more than 90% of the fire and with an average of only 8.5 dead trees. It has to be taken into account that the agents were moving random and were not aware of where the fire was to just head there and eliminate it. Almost half of the forest caught fire and about 95% were controlled by the agents. This shows that with minimalistic damage, the forest can be saved with 40 agents. |
| Experiment 4 | **6.2** | **0.6** | **393.2** | In this scenario, the fire started with one tree and there were all 40 units there to save it, as it can be seen the average dead trees was less than 1 when atleast 7 trees were on fire from 400. This shows that with 40 agents, even if a fire starts with one tree, can be controlled before it spreads through the forest and most of the forest remains safe. |
| Experiment 5 | **3.2** | **6.7** | **390.1** | In this scenario, there were only 10 units to extinguish fire instead of 40 as from experiment 4 and this would seem more realistic as 10 units can be deployed in each forest and as it can be seen with the results, most of the forest remains safe from fire and only 10 trees caught fire from which 30% were saved with the minimum resources available, however this shows that even with putting out fire on limited trees prevented the fire going wild all over the forest. |

**The Individual experiment data is shown on the next page**



# Design Choice to increase Efficiency

**Design Choice:**

The design choice when tested with the set of experiments was not quite accurate as the more units of water loaded onto the agent made it slower and it was not able to reach further from the base and hence the number of dead trees increased. Below are the set of experiments conducted to test the wise choice.

**Experiment: 6:** Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:40

Initial-water: 50

This experiment was carried out to see whether the wise decision of refueling maximum number of units hold or not and the results are discussed below in detail, however the initial-water was set to 50 as it was the maximum possible and all other parameters were set to their maximum as the focus was to see how much initial-water affects the dead trees. This should be kept in mind that the more the water, the slower the unit gets and hence it wont be able to reach the corners in time to save the trees with carrying max number of water units.

**Experiment: 7:** Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:40

Intital-water:25

This experiment was carried out to check whether halving the maximum number of units result in fewer dead trees. As the agents are now carrying half the weight, they should be able to reach the corners with still having some water. The purpose of this experiment was to check how the agents will behave with having half water supplies with respect to maximum capacity.

**Experiment: 8:** Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:40

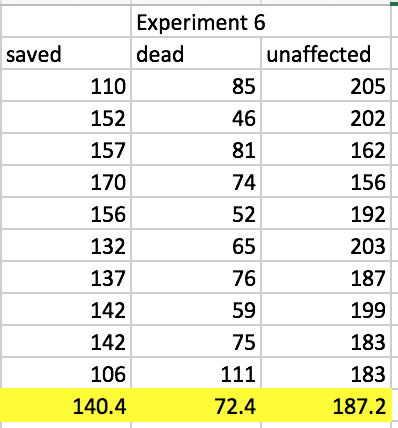
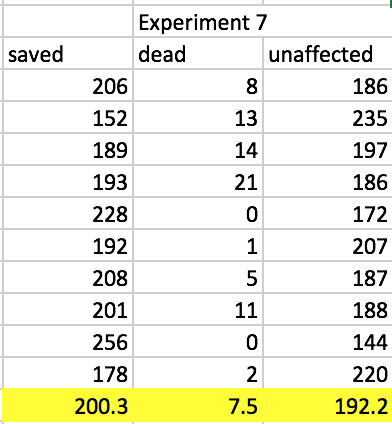
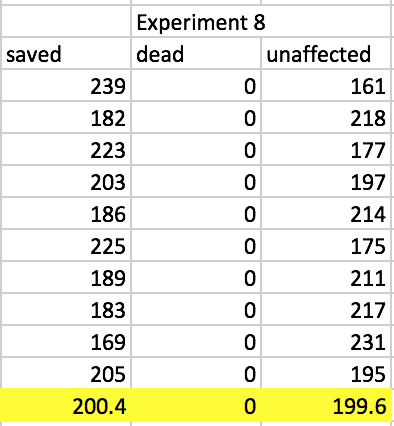
Intital-water:10

This experiment was carried out to check how the agent behaves when it has only 10 units of water as opposed to earlier experiment with 25 water units. The number of fires and number of units were deliberately made to be the same to have a fairer comparison. The results of the experiment are shown below:

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Saved trees | Dead Trees | Unaffected Trees | Analyses |
| Experiment 6 | **140.4** | **72.4** | **187.2** | The theory was put to test in this experiment as it was tested to refuel to maximum capacity every time an agent runs out of fuel. The results show that the number of dead trees were extremely high because the speed of the units got so slow that they could not reach the end trees in time to put out the fire. The agents were only able to extinguish about 60% of the fire and the rest led to dead trees. |
| Experiment 7 | **200.3** | **7.5** | **192.2** | In this experiment, the initial-water was halved and the results were astonishing, the average number of dead trees dropped from 72.4 to 7.5. This is a huge decrease and it shows that our initial choice was not accurate as the more units of water the agents carried, the more the dead trees were. |
| Experiment 8 | **200.4** | **0** | **199.6** | Lastly, the initial-water was even reduced to 10 and the results were unbelievable as in the 10 runs, the number of dead trees were 0, to check this theory even further, some tests were done unrecorded and results were the same, the number of dead trees remained to be 0 in all of the tests performs. This concludes that 10 is perhaps the magic number for forest fire case and this also proves the initial choice was wrong as the more units of water the agents carry, the slower they get and hence there is an incredible amount of increase in number of dead trees. |

The individual test results are shown below:



# 

# Improvements

There are a number of improvements which can be suggested to the current system as in the current system, the agents just move randomly which is not efficient as it may work sometimes and may not work at all at other points. Below are the detailed points of improvements with their justifications of proposal:

1. **Neighbor trees** – If a tree is on fire then the agent should check its neighboring trees for any traces of fire because its very likely that if one tree is on fire then its neighbor may also be on fire and if that is the case then agent checking for that will increase efficiency by eliminating fire which comes in the form of cluster and stop the fire early on as one cluster of trees on fire will not affect the others.
2. **Multi-agent communication** – agents should help each other in terms of communication, right now the agents move randomly without any knowledgebase whatsoever, even if one agent has seen fire and did not have enough water to extinguish it, it does not let other agents know that there is a fire at these coordinates so maybe some other agent around could put it out, this is very similar to how the police coordinates when someone is running away and nearest police car reacts accordingly, this will still be using the subsumption architecture but will help to increase the efficiency as if one agent cant eliminate the fire then rest will have a goal to reach and they will mostly reach and extinguish the fire before the agent which reported it comes back and this way they can all delegate the work and efficiency will be increased rapidly as they are no longer moving around randomly but they rather have a goal to reach.
3. **Water lending between agents** – One of the other improvements which could be suggested is the water lending between agents as for a scenario where one agent just came fully refueled, and its close by agents are running out of water, instead of other agent going back to base to get more water, the agent with full water can lend half of its water to the other agent and they can extinguish more fires together and this will help to increase efficiency as by giving half of the water away, the agents speed is doubled and with two agents instead of one, they can now extinguish more fires then one could alone.

# Advantages and Disadvantages

There are many advantages of the reactive agent approach in the forest fire problem as the forest fire happen all of a sudden and it needs reactive agents to handle the fire immediately, it cannot be proactive because no one can tell when a fire will start hence there are several benefits of reactive agent approach. For example the reactive agent will extinguish fire as soon as it detects it but if its not around then it will not help.

Lets take a look into some disadvantages of the reactive approach first:

One of the disadvantages of the reactive approach is that it does not care about the bigger picture, it will only take out the fire it detects first rather than how proactive approach can generate an effective plan to reduce the number of dead trees based on trees on fire and then give agents only the number of water units required by them to eliminate the fire and this way some of the agents can remain idle at the base and be dispatched with a plan once a fire is known, this will be more like an ambulance rescue service which is highly effective.

There are advantages and disadvantages to the reactive approach and in my opinion, the hybrid of both proactive and reactive approach will solve the problem in the best manner possible, however evaluating reactive approach based on the forest fire problem, It works reasonably well as experiments confirm.