|  |
| --- |
|  |

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.

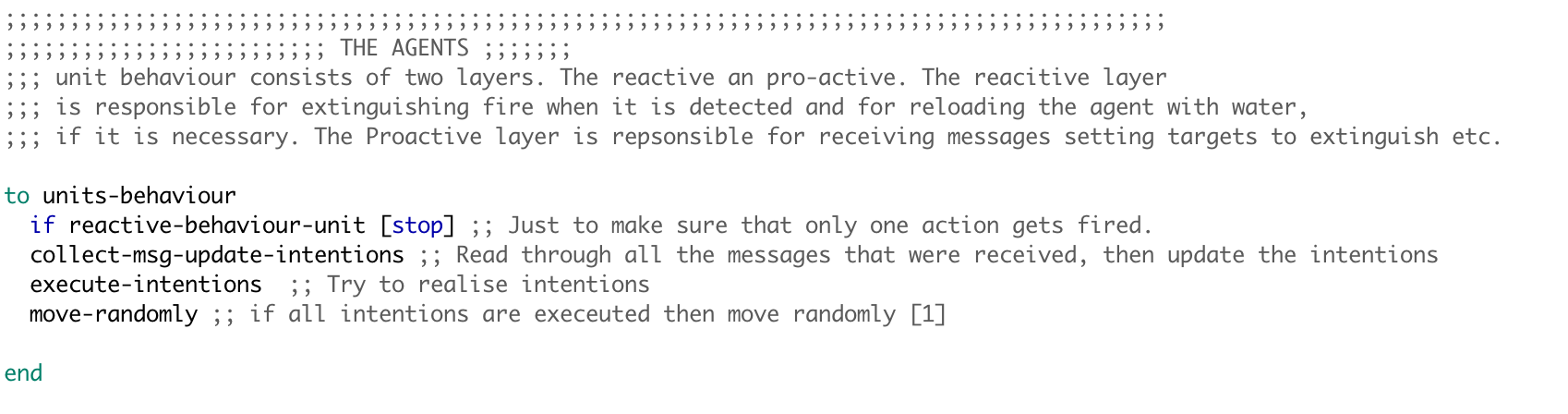


Figure Showing the addition of calling move-randomly to make ground units move randomly

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]

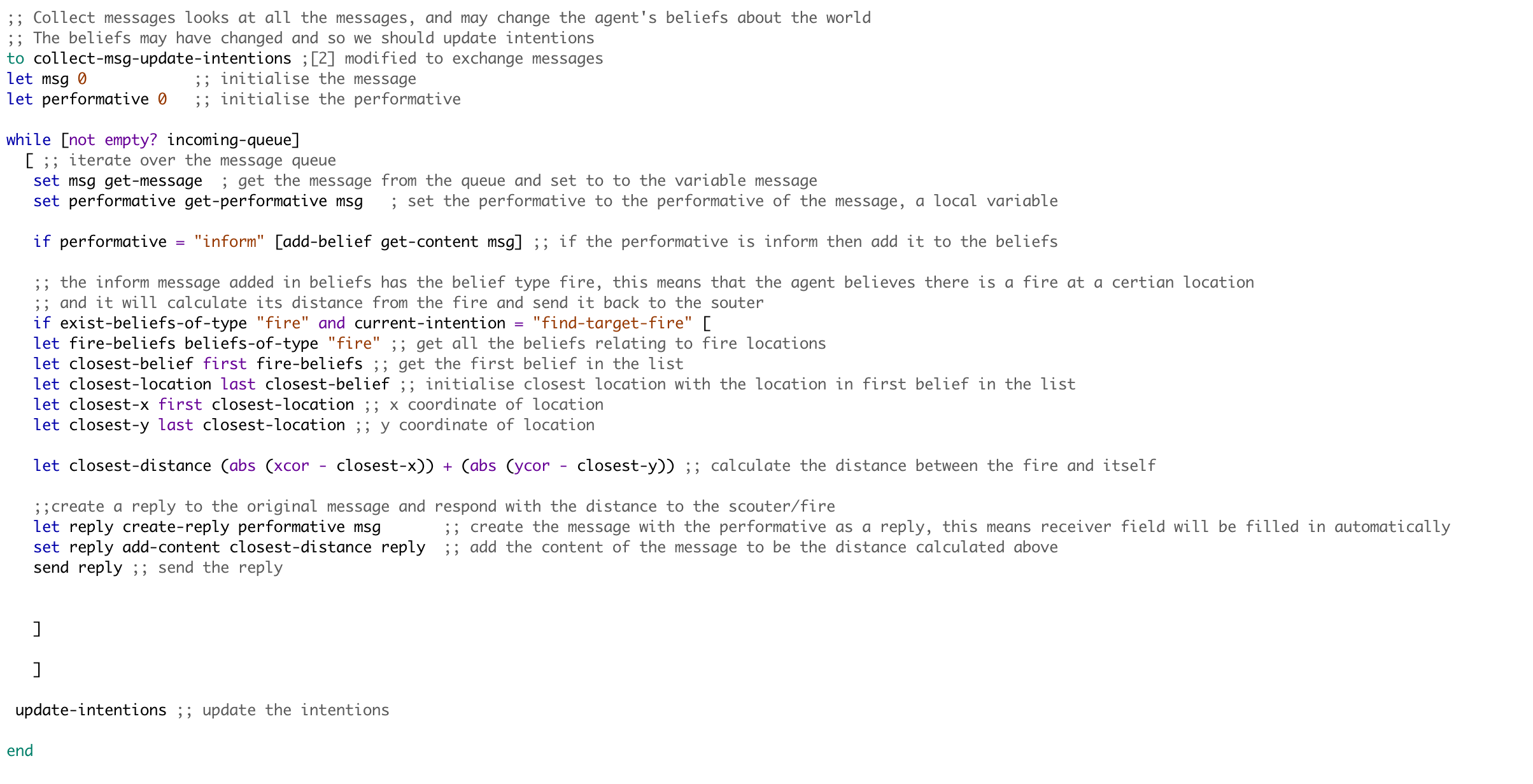


Figure Showing replying of the broadcast message with their distance

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]

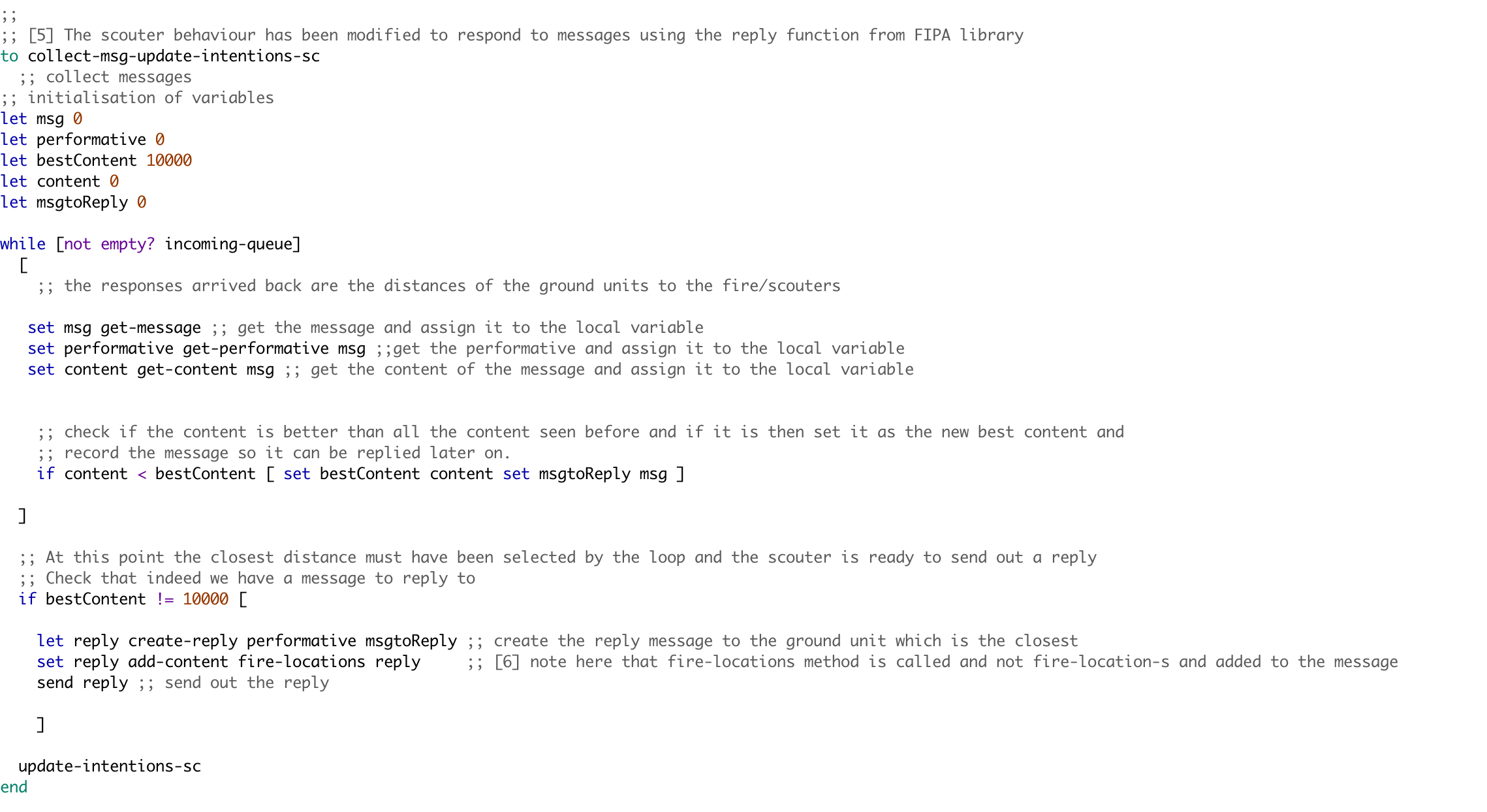


Figure Showing the altered code in collecting message and updating intentions of the scouters

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]

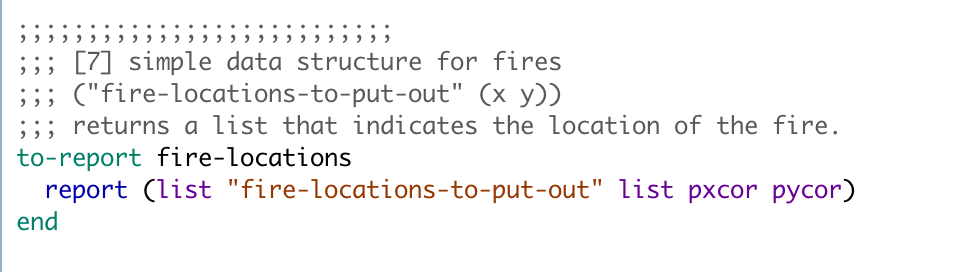


Figure Showing a new report mechanism added to aid the BDI design

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].

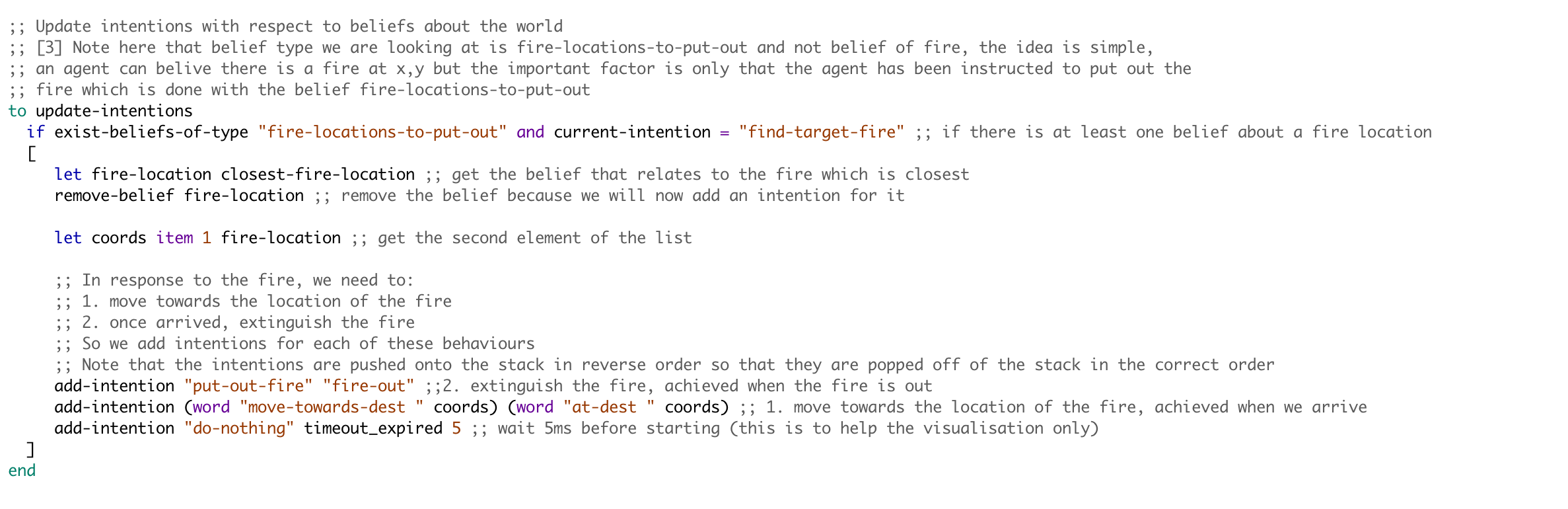


Figure Showing the alterations in the update intentions of the ground units

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.

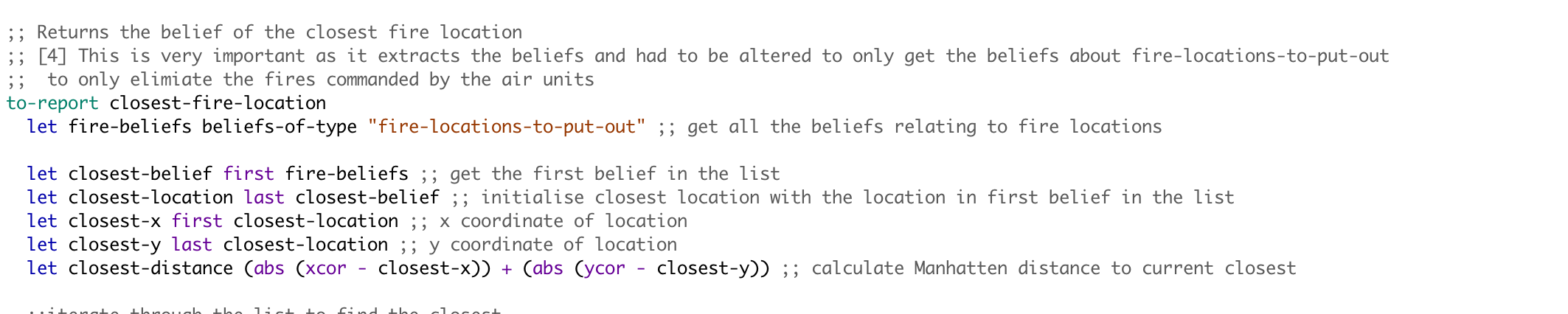


Figure Showing alterations in the closest-fire-location procedure

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.

# Experiments

2 **Description and Justification of the experiments:**

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 Scouter-num:1

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 Scouter-num:1

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: 500 Number-of-fires:40 Scouter-num:30

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 Scouter-num:30

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: 500 Number-of-fires:20 Scouter-num:10

**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**.

**Experiment: 6:**

Fire-unit-nums: 10 Tree-num: 200 Number-of-fires:20 Scouter-num:5

**Experiment: 7:**

Fire-unit-nums: 30 Tree-num: 300 Number-of-fires:20 Scouter-num:20

**Experiment: 7:**

Fire-unit-nums: 40 Tree-num: 500 Number-of-fires:40 Scouter-num:25

10 200 20 5

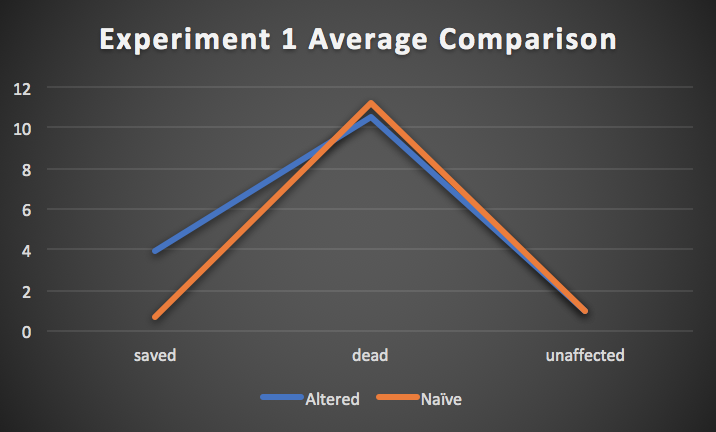
**Results:**

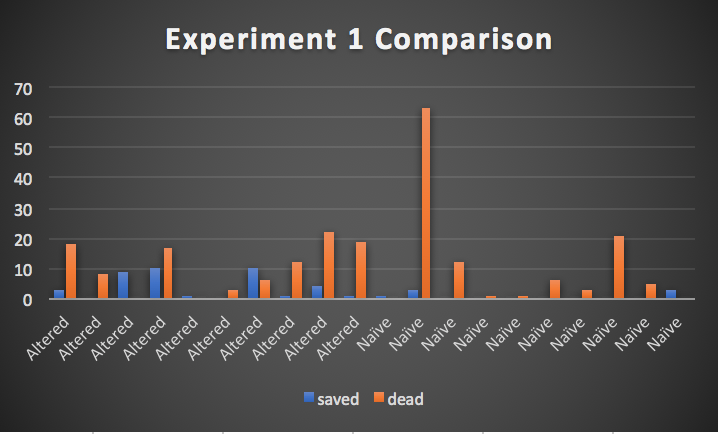
The results are an average of running the simulation 10 times to get a better estimate rather than running it a single time. (The individual results are shown below)

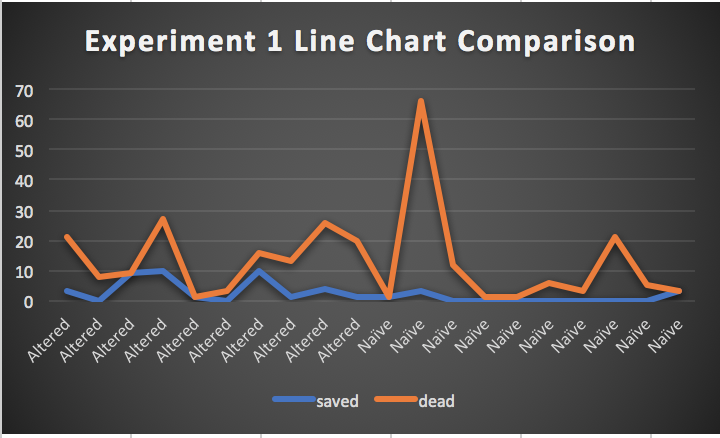
|  |  |  |  |
| --- | --- | --- | --- |
|  | Saved trees | Dead Trees | Unaffected Trees |
| Experiment 1 | **3.9/0.7** | **10.5/11.2** | **385.6/388.1** |
| Experiment 2 | **0.6/0.2** | **1.1/1.6** | **98.3/98.2** |
| Experiment 3 | **326.7/53** | **0/329.1** | **173.3/117.9** |
| Experiment 4 | **8.1/7.4** | **0/2.9** | **391.9/389.7** |
| Experiment 5 | **242.2/59.2** | **16.9/183** | **240.9/257.8** |
| Experiment 6 | **49.2/22.3** | **0.7/45.6** | **150.1/224.5** |
| Experiment 7 | **74.1/29.2** | **0/45.6** | **225.9/224.5** |
| Experiment 8 | **316.6/61.2** | **0/320.2** | **183.4/118.6** |

**Experiment: 1:**

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



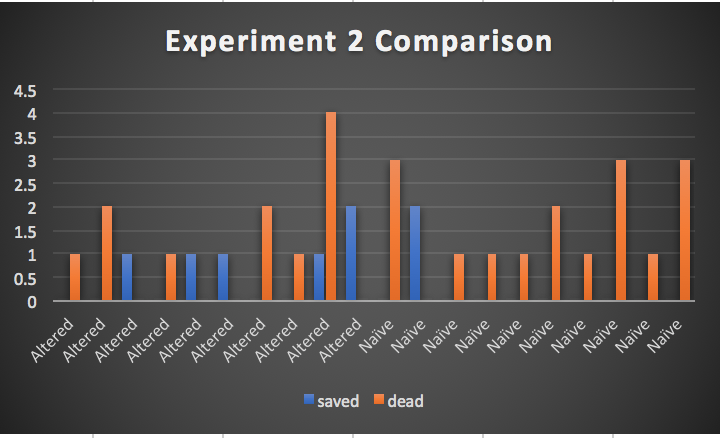


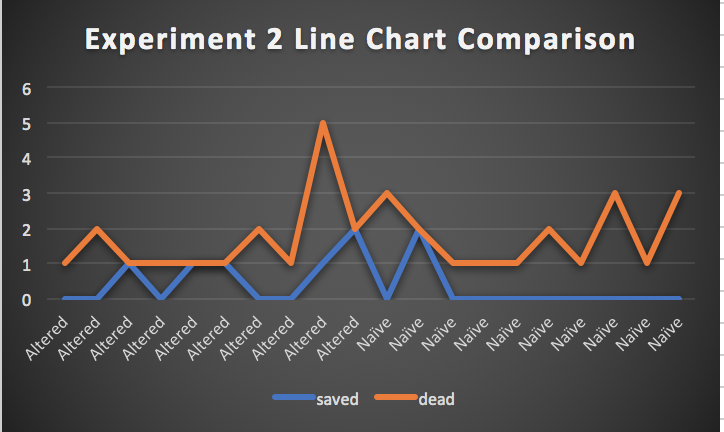


**Experiment: 2:**

Fire-unit-nums: 1 Tree-num: 100 Number-of-fires:40 Scouter-num:1

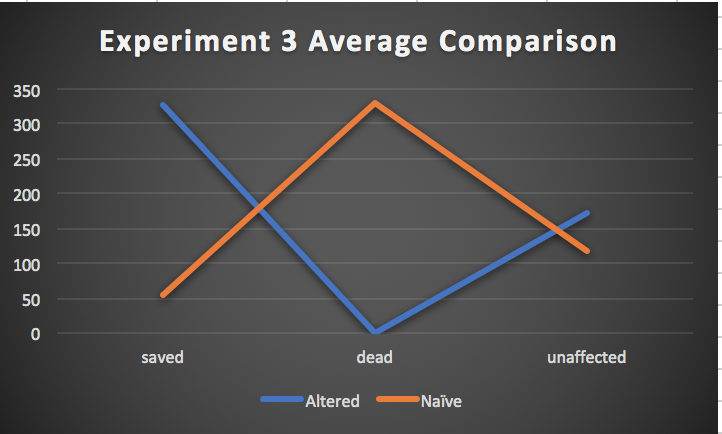


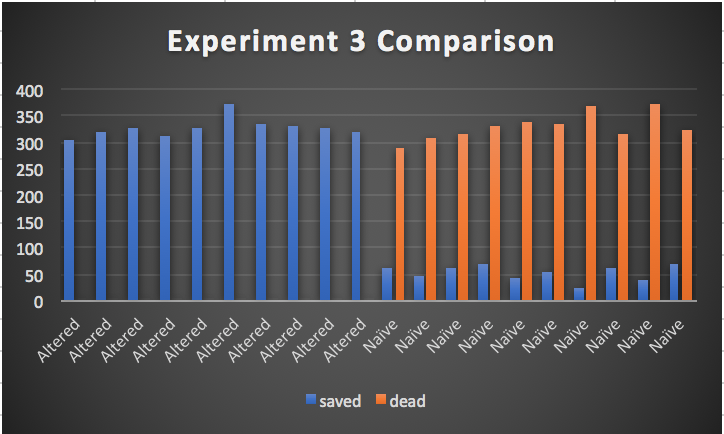


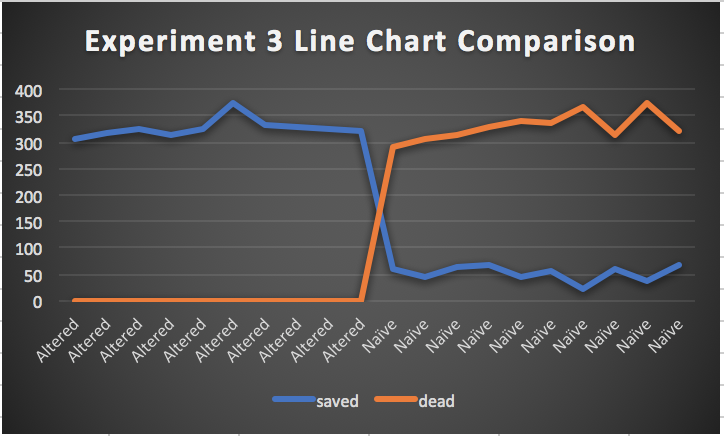


**Experiment: 3:**

Fire-unit-nums: 40 Tree-num: 500 Number-of-fires:40 Scouter-num:30

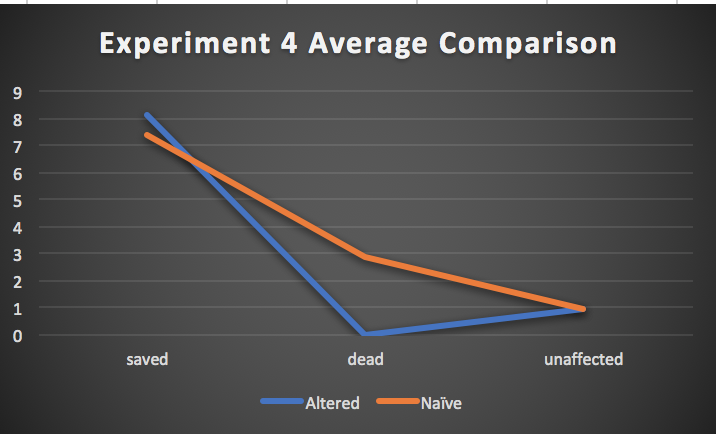


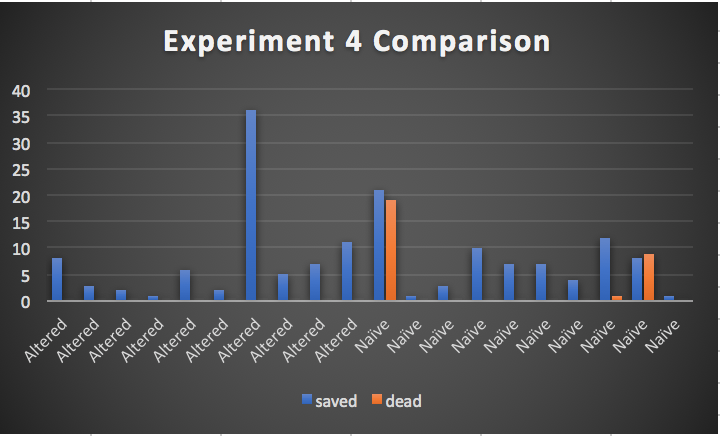


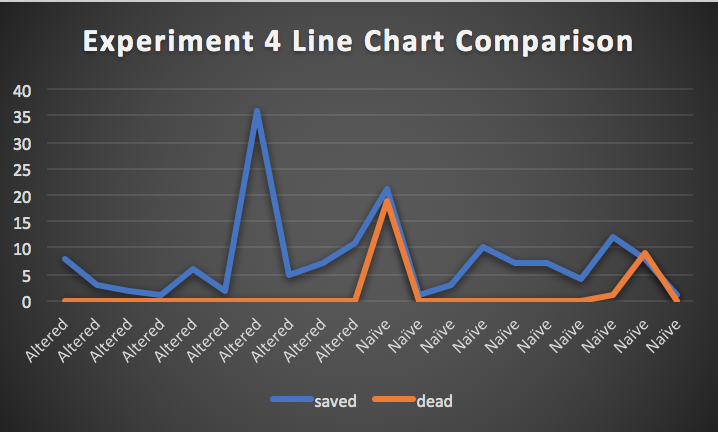


**Experiment: 4:**

Fire-unit-nums: 40 Tree-num: 400 Number-of-fires:1 Scouter-num:30

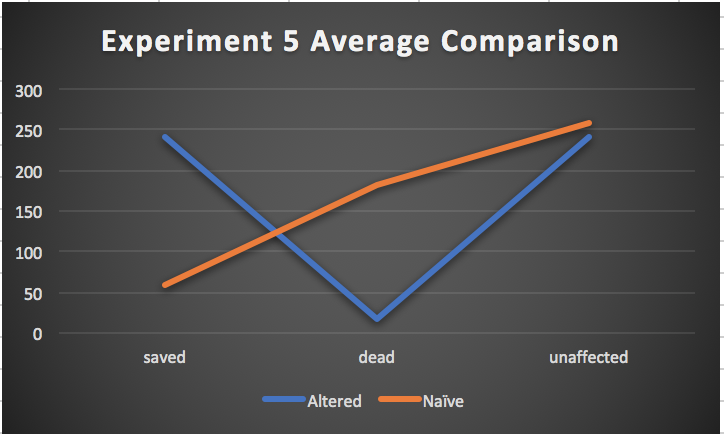


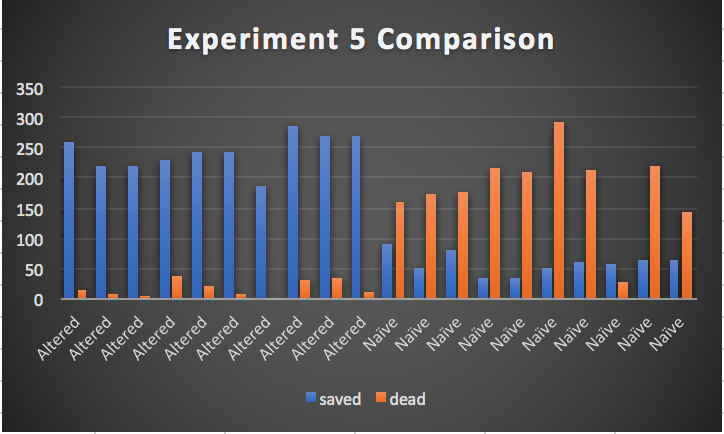


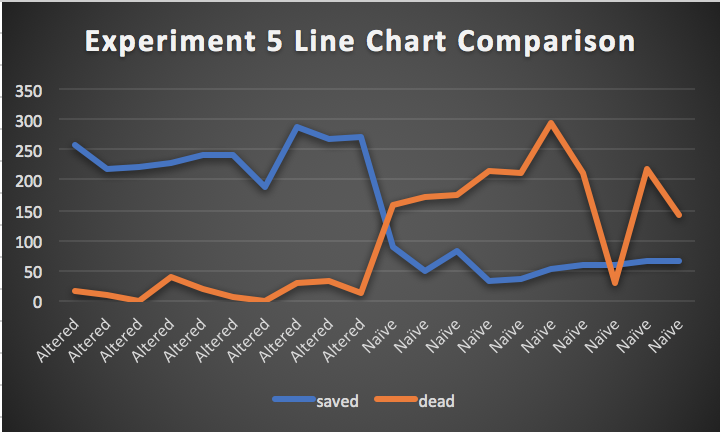


**Experiment: 5:**

Fire-unit-nums: 10 Tree-num: 500 Number-of-fires:20 Scouter-num:10

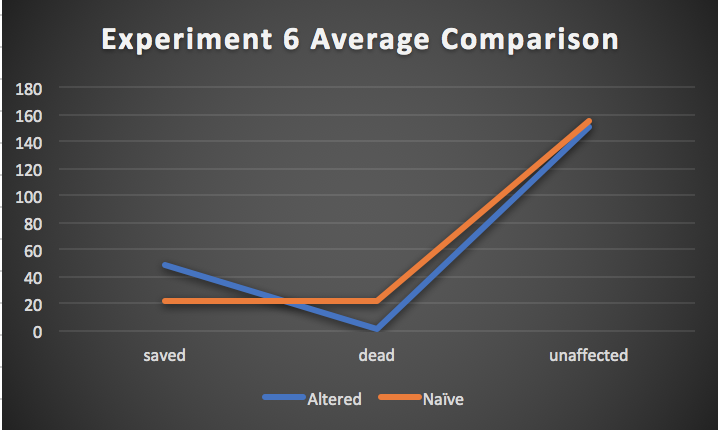






**Experiment: 6:**

Fire-unit-nums: 10 Tree-num: 200 Number-of-fires:20 Scouter-num:5



**Experiment: 7:**

Fire-unit-nums: 30 Tree-num: 300 Number-of-fires:20 Scouter-num:20

**Experiment: 7:**

Fire-unit-nums: 40 Tree-num: 500 Number-of-fires:40 Scouter-num:25

# Advantages

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.

# Appendix

# ../../Desktop/Screen%20Shot%202017-12-06%20at%201.10.42%20A

