Strategies for an Efficient Response to Wildfires

Autonomous Agents and Multi-Agent Systems

Guilherme Gonçalves
Instituto Superior Técnico
ist195585

Marina Gomes
Instituto Superior Técnico
ist195637

Martim Santos
Instituto Superior Técnico
ist195638

ABSTRACT

The aim of this project is to study efficient combat wildfires strategies by leveraging the collaborative efforts of the involved agents. Our system will resort to model and study firetrucks and helicopters that possess varying behaviors, including coordinated and cooperating behaviours capable of deciding what is the best course of action to extinguish all active wildfires, taking the least amount of time possible and using the least amount of resources.

KEYWORDS

Multi-Agent System; Cooperation; Coordination; Reasoning; Decision Making; Search; Algorithms; Heuristics; Performance; Efficiency

1 INTRODUCTION

1.1 Motivation

Wildfires are a very common occurrence in warm countries, like Portugal, especially in the summer season. Due to its strong destructive power, there's an urge to extinguish them as soon as possible, which requires swift and intelligent resource allocation and management. In order to combat these adversities, there is a need for coordinated efforts between various agents and resources. By addressing these challenges, it is possible to reduce the damage caused by wildfires and ensure the safety of both people and the environment.

Our goal is to simulate a Multi-agent system that mimics potential wildfire scenarios and examine how various approaches impact the time required to extinguish wildfires and the utilization of resources, namely water. Consequently, we intend to investigate the influence of various agent types, as well as coordination and cooperation mechanisms, on the duration needed to effectively suppress all fires.

1.2 Problem Definition

1.2.1 Fire trucks & Helicopters.
Fire trucks and Helicopters have a shared common goal of extinguishing all existing wildfires as quickly as possible.

1.2.2 Water Sources.
Water sources provide water to the Fire trucks' and Helicopters' tanks.

1.2.3 Wildfires.
Wildfires emerge in the environment and can become increasingly difficult to put out, necessitating more water.

1.3 Implementation

Regarding the actual implementation, we decided to use *Python* [4]. Additionally, we utilized some of its libraries (like *NumPy* [3], *scikit-learn* [5], etc.) since they provide powerful data structures, mathematical complex operations and machine learning algorithms to implement the agents. To compare results, evaluate performances and generate plots, we decided to use *Matplotlib* [2] since it is a data visualization library for creating static, animated, and interactive visualizations.

For the environment, we used a level-based Foraging (*LBF*) [1] environment due to the fact that it provides a well defined multi-agent reinforcement learning environment.

The object-oriented programming approach was employed to represent agents, with their internal states and sensors being denoted by attributes and methods. In other words, we employed multiple classes and subclasses to effectively represent the various types of agents.

2 SYSTEM ARCHITECTURE

2.1 Environment

The environment is represented by a grid $X \times Y$. The positions of the agents, water sources and wildfires are denoted by (x, y) coordinate pairs and their direction is determined by their direction sensor, if applicable.

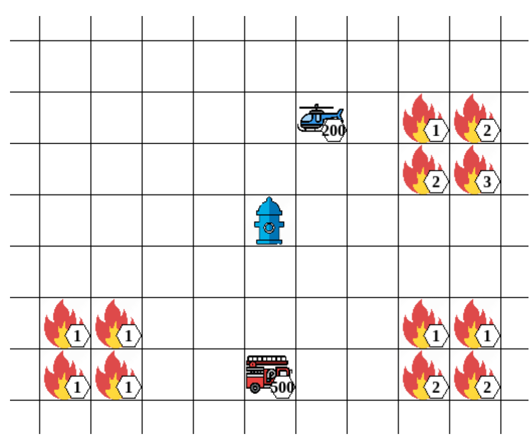

Figure 1: Fragment of a state of the Environment with three active wildfires, two different Agents and a Water Source

AASMA Project, IST, 2023, Lisbon, Portugal

https://doi.org/

Each wildfire will occupies a fixed number of tiles on the grid, with varying intensity in each tile. A tile can be seen as a fire front. The intensity of fires in tiles may either remain constant or increase after a certain number of time steps until reaching a maximum level.

Our objective is to replicate intricate real-life scenarios, capturing the nuanced dynamics and complexities, in order to create a simulation that closely emulates the conditions and challenges encountered in actual firefighting operations.

The environment is:

- **Accessible**: The agents are able to obtain a complete perception of the current state of the environment.
- **Deterministic**: Each action has a single guaranteed effect.
- **Dynamic**: Wildfires intensity might increase while an agent is updating its internal state.
- **Discrete**: There is a finite number of possible actions and sensors for the agents.

2.2 Multi-Agent System

We defined the following types of agents for this problem:

- *Fire Trucks*: are equipped with larger water tanks, reaching the capacity of 500 liters. Additionally, they must execute a turn before altering their direction of movement.
- *Helicopters*: have smaller water tanks, with maximum capacity of 200 liters. *Helicopters* have the ability to move in any direction without the need to turn.

The agents have the ability to extinguish fires by utilizing the water in their tanks, either to fully extinguish the fire or to reduce the intensity of its fronts. When an agent's tank is depleted and there are still fires present in the environment, the agents must replenish their tanks in order to continue with their extinguishing tasks. In a scenario involving a total of n_agents agents, $\lfloor \frac{n\_agents}{2} \rfloor$ of them are designated as *Helicopters*, while the remaining $n\_agents - \lfloor \frac{n\_agents}{2} \rfloor$ agents are *Fire Trucks*. Therefore, in the case of an odd number of agents, there will always be one additional *Fire Truck* compared to the number of *Helicopters*.

2.2.1 *Properties.*

In terms of agent properties, we have:

- **Reactivity**: Agents react to changes of the environment, namely aggravations of wildfires.
- **Proactiveness**: Agents are able to identify and react to opportunities that arise (e.g. wildfire aggravate and the agent does not have sufficient water to extinguish it, it can change its target)
- **Social Ability**: Fire trucks and Helicopters cooperate and coordinate themselves to reach their objective.
- **Rationality**: Agents can act in a way that minimizes the total time.

It is important to note that the agents' properties depend on the heuristics used, which will be addressed later.

2.3 Sensors and actuators

The agents are equipped with both fire and GPS sensors. *Fire Trucks* will additionally have direction sensors. These sensors will enable the agents to detect changes in the environment such as new wildfires or existing ones that aggravated as well as their moving direction. Lastly, to fully replicate real-life firefighting scenarios, the GPS will furnish crucial details on the location and orientation of the other agents, enabling the communication between them and fostering cooperation to achieve their desired common goal.

Agents	Sensors	Actuators
Helicopter	_active_fires _closest_fire _order_closest_fires _strongest_front_closest_fire _closest_water_source	move extinguish refill
Fire Truck	_active_fires _closest_fire _order_closest_fires _strongest_front_closest_fire _closest_water_source	move turn extinguish refill

Table 1: Agent Sensors and Actuators

2.4 Cooperation and Coordination

Given the lack of prior knowledge regarding the intended directions and objectives of each agent (specifically, which fires to tackle), coordination and cooperation amongst the agents become imperative in order to strive for better results. An exemplification of this can be witnessed in the following scenario: Due to the intended presence of enough agents at a particular fire, one or more agents may decide to shift their focus to another fire(s), leading to changes in their actions and hopefully reducing the time taken to extinguish the fires.

2.5 Fires

The *Fires* occupy a predetermined number of tiles on the field, referred to as *fronts*. Each front is associated with a level that represents the intensity of the flames at its specific position. The overall level of a fire is determined by the cumulative sum of the levels across all fronts. Over time, fires can intensify, causing the levels of the fronts to increase after every n time steps. The ultimate objective is for the agents to extinguish all fires in order to achieve the goal.

2.6 Water Source

The *Water Sources* will allow the trucks to refill their water tanks, since putting out fires will deplete it. If an agent exhausts its water supply, it cannot combat the wildfires until it refills its water tank. The number of *Water Sources* will always be $\lfloor \frac{fires}{2} \rfloor$, where $fires$ represents the total number of fires.

2.7 Agent Behavior

The agents (both *Fire Trucks* and *Helicopters*) will use different methodologies (in another words, heuristics) to choose their next action. In our implementation, we provided many possible examples of different approaches, in order to conclude which are the best ones.

2.7.1 *Random and Pseudo Random Agents.*

Starting with the *Random Agents*, these agents take no consideration for the observations provided by the environment and choose their next action randomly (with equal probability on each action). We also tried to improve random agents with *Pseudo Random Agents*. These agents do not have equal probabilities of choosing their next actions. For instance, actions like doing nothing or turn have lower probabilities than move, refill or extinguish. It is still worth mentioned that neither of these agents will try to move towards a Water Source in case they run out of water. *Random* and *Pseudo Random Agents* will be unable to continue extinguishing fires, unless their able to reach a Water Source.

2.7.2 *Heuristic Agents.*

Then, we moved on to slightly more intelligent agents, the Heuristic Agents. These agents will be able to choose their next fire based on different heuristics. Also, should the agent find itself without water, it will move towards refilling its water. The heuristics implemented are:

- **Closest Fire**: The agent will prioritize the closest fire and will behave in order to extinguish it.
- **Strongest Front of the closest Fire**: The agent will prioritize the strongest front (i.e the tile with the highest level) of the closest fire and will behave in order to extinguish it.
- **Closest Front that it can put out**: The agent prioritize the closest front from any fire that it can put out completely in one extinguish action (in another words, tiles such that the fire level isn't bigger than agent's water capacity). If no fires are available, the agent will remain inactive and stationary, without taking any action.
- **Strongest Fire**: The agent will prioritize the strongest fire burning in the grid.
- **Weakest Fire**: The agent will prioritize the weakest fire burning in the grid.

While the above agents are more objective driven than the random agents, they lack coordination. For that reason, we also proposed the implementation of agents with coordination to address our problem. For the coordination, the Nash equilibrium will be achieved by either using social conventions or a role-based decision making system. In both cases, many different methodologies were explored, tested and evaluated.

2.7.3 *Coordination with Social Conventions.*

For coordination using social conventions, we define a unique order for both agents and fires. Each agent chooses a specific fire to go to using that order according to the following conditions:

- The index of the agent in the agents' convention corresponds to the index of its assigned fire in the fires' convention.
- If the number of agents exceeds the number of fires, each agent is assigned a fire that corresponds to the fire with an index equal to the remainder obtained by dividing the agent's index in the agents' conventions by the number of fires.

The implementation has two variants in terms of when the social conventions are computed. In the first variant, social conventions are computed just once, in the first step. The second variant, computes the social convention at each step, updating the order of agents and fires. For evaluation purposes we implemented different types of social conventions, namely:

- Agents and fires ordered by descending order of level.
- Agents and fires ordered by default id.
- Agents ordered by descending order of level and fires ordered by ascending order of level.

2.7.4 *Role-based Coordination.*

For the role-based coordination, a role equals a fire (i.e. having role 0 means that agent will be in charge of fire 0). As long as the water assign to that fire (in another words, the total sum of the remaining water in the fire's assigned agents) doesn't exceed the total fire level, agents are allowed to be assigned to a fire. In the event of every fire has enough water to be put out, the unnecessary agents will be distributed evenly throughout all the fires (in an effort to speed up the process). The implemented potential functions are as follows:

- **Distance to the closest fire tile**: The Manhattan distance to all the fire's tiles will be computed. Finally, the potential value will be the negative of the smallest value computed. Meaning for fire f_i, the potential function is then given by

$$\mathcal{P}(a, f_i) = -\min_{\forall t \in f_i} \text{Manhattan\_distance}(a, t)$$

- **Distance to the closest fire tile and water remaining**: The Manhattan distance to all the fire's tiles will be computed. Finally, the potential value will be the subtraction of the remaining water and the smallest value computed. Meaning for fire f_i, the potential function will be the following formula

$$\mathcal{P}(a, f_i) = water - \min_{\forall t \in f_i} \text{Manhattan\_distance}(a, t)$$

- **Distance to the closest fire tile and faction of water remaining**: The Manhattan distance from agent a to all the tiles t from fire f_i will be computed. Finally, the potential value will be influenced by the fraction of remaining water and the smallest distance value computed. The potential function is then given by

$$\mathcal{P}(a, f_i) = \frac{water}{water\_capacity} - \min_{\forall t \in f_i} \text{Manhattan\_distance}(a, t)$$

3 EVALUATION

3.1 Time

Time (i.e the required time steps to put out all fires) is the the main focus of this system, since fire that burn the longest cause the most damage and can be harder to extinguish. For that reason, this will allow us to evaluate the system, given that the best approaches efficiently extinguish all the wildfires in the least amount of time steps.

3.2 Water Usage

Water usage by each agent can also give us insight about the efficiency of the system. It can be related to time, since agents need to travel to the Water Source to refill their water tanks. The agents might make decisions regarding whether to go fill their water tanks, thereby minimizing the number of trips required to the Water Source in the long run, or to opt to prioritize fires and only go refill when strictly needed.

3.3 Comparing Execution Modes

In order to comprehensively evaluate the system across various modes, we mainly employed the metric of *Time*. Furthermore, to conduct a more extensive analysis of the system, we made adjustments to multiple parameters, including the number of fires (*--fires*), number of agents (*--n_agents*), maximum number of steps (*--max_steps*), mode selection (*--mode*), and steps used to increment the fire level (*--steps_incr*) and a seed to guarantee the comparation between equal environments (*--seed*).

We used the following values for the flags to compare the different agents:

- *--mode*: [0, 1, 2, 3] = [Random vs Pseudorandom, Greedy, Social Conventions, Roles Based]
- *--times*: [10, 100, 500, 1000]
- *--fires*: [3, 5]
- *--n_agents*: [2, 7]
- *--max_steps*: 400 (for mode 0, we also experimented 1000)
- *--steps_incr*: [None, 50]
- *--seed*: [False, True]

3.4 Random and Pseudo Random Agents

Impact of Randomness: As expected, these agents face two main challenges. Firstly, they either require the maximum number of steps to fully extinguish all fires or fail to achieve the goal entirely. This is primarily due to their lack of utilizing observations to assess their capability to extinguish a fire, resulting in potentially inefficient movements (even when they are adjacent to a fire). Secondly, these agents are further hindered by the fact that if they run out of water, they become ineffective, exacerbating their existing issues.

Modified Action Probabilities: In Figure 2, a noticeable improvement of the *Pseudo Random Agents* compared to the fully *Random Agents* is observed. This outcome highlights the impact of distinct action probabilities on the performance of the *Pseudo Random* agents, enabling them to surpass the capabilities of the *Random* agents in specific environments. However, it should be noted that this improvement may be directly linked to the relative location of each agent with respect to the fires.

Favorable Scenarios: To improve the performance of these agents and potentially reduce the number of steps required, several strategies can be implemented. One approach is to increase the quantity of *Fires* and *Water Sources* within the environment. By doing so, more opportunities are created for *Refill* and *Extinguish* actions, enabling the agents to reach their goal more swiftly. Moreover, increasing the number of agents also enhances the likelihood of successfully extinguishing fires, resulting in an overall improvement in performance.

Relevance of a Seed: Lastly, it is important to acknowledge that the selection of a poor seed for the random action generator can significantly affect the performance of these agents. In a scenario where an unfavorable seed consistently prompts the agent to select

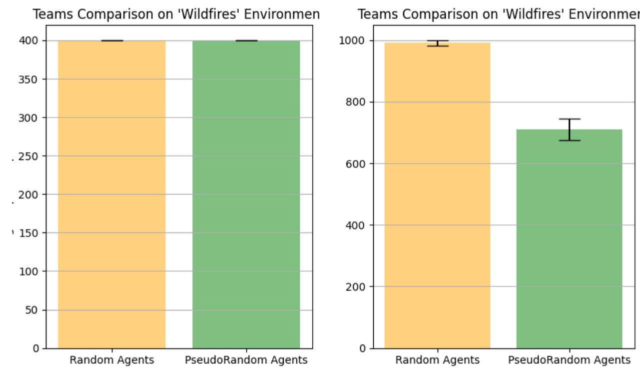

Figure 2: Random vs Pseudo Random results for fires=5, agents=7, max_steps=[400, 1000]

the same unfavorable action, progress towards the global goal is impeded, leading to a state of stagnation.

3.5 Greedy Agents

In the evaluation of the Greedy agent heuristics, we have made several significant observations:

Challenges with Increasing Fire Intensity: As anticipated, the Greedy heuristics, which focuses on targeting the closest fire that it can fully extinguish, encounters difficulties when the fire intensity increases beyond a the agent's capability. In such cases, the agents may become immobilized, unable to take any action due to the absence of fires that they can fully extinguish as it can be seen in 3.

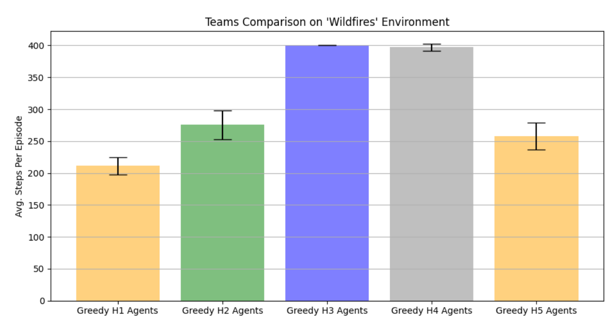

Figure 3: Results for mode=1, times=10, fires=3, agents=2, seed=816120, steps_incr=30

Effect of Number of Fires with Constant Agent Count: When considering the scenario of a fixed number of agents with varying numbers of fires, we found that the overall performance remains relatively stable. The distribution of results remains largely unchanged, with only a marginal increase observed in the mean number of steps required to complete the task, as it can be seen in 4. This observation can be attributed to the availability of additional water sources to combat the increased number of fires.

Impact of Increased Fires and Agents: Scaling up both the number of fires and the number of agents does not significantly

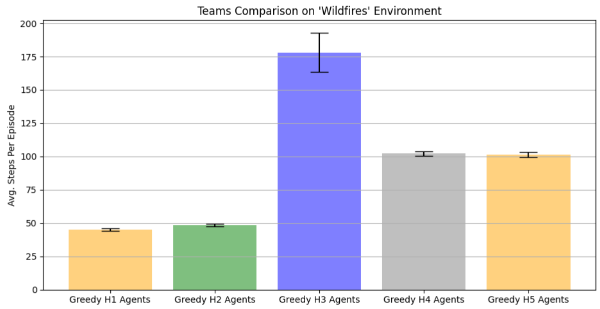

Figure 4: Results for mode=1, times=500, fires=5, agents=7, seed=None, steps_incr=None

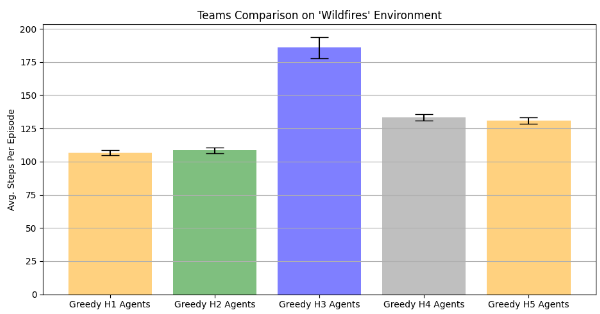

Figure 5: Results for times=1000, fires=3, agents=2, seed=None, steps_incr=None

alter the conclusions drawn from our analysis. While there may be slight variations observed when examining a larger number of trials, the overall patterns and trends remain highly consistent. As expected, when the number of agents is significantly higher than the number of fires, the mean time steps to reach the objective become lower. It is important to note that as long as the proportion of fires to agents remains relatively balanced, the impact on the results is minimal.

Analysis of Execution Times: Our evaluation also considered the influence of execution times on performance. We observed that certain scenarios exhibited higher variance in performance for few execution times (Figure 6), prompting us to conduct additional experiments using larger values (e.g., 500 and 1000 times) for deeper insights. Interestingly, even for longer time intervals, the Greedy heuristic H3 consistently exhibited inferior performance compared to other strategies (Figures 4 and 5).

Effect of Agent Count: The number of agents deployed has a significant impact on the time required to complete the task. As expected, increasing the number of agents leads to reduced completion times, as more agents are available to combat the fires simultaneously.

Role of Seed in Results Consistency: To explore the role of initial conditions, specifically the agents' starting positions in relation to the fires, we conducted experiments with fixed seeds. We anticipated that results could potentially be influenced by these initial conditions. However, our analysis revealed that, in the majority of cases, fixing the seed resulted in reduced variance in outcomes, with overall relative performance remaining consistently across trials, although there are a few exceptional cases. Figures 3 and 6 provide visual evidence of this phenomenon.

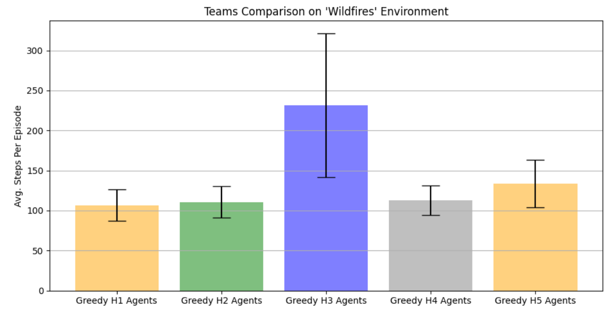

Figure 6: Results for times=10, fires=3, agents=2, seed=None, steps_incr=None

3.6 Social Conventions

For the evaluation of Social Conventions, we compared the performance between the different social conventions implemented, along with the different ways of updating those conventions.

Impact of Conventions Update: The Social Convention agent C4 dynamically updates its conventions at each step to adapt to the constant changing levels of both agents and fires. However, this constant updating of conventions leads to frequent changes in the agents' targets, resulting in inefficient use of steps. As depicted in Figure 7, the performance of this Social Convention agent is notably inferior comparing with other agents.

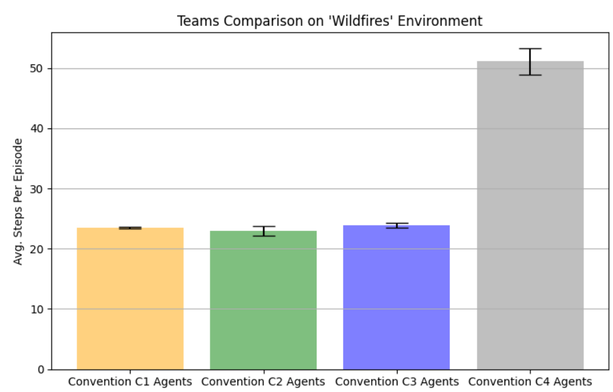

Figure 7: Results for times=100, fires=3, agents=7, seed=74831, steps_incr=50

Impact of Fire Level Increment: By incrementing the fire level after a specified number of steps we intensify the difficulty of extinguishing the fires. The approach on Social Convention C3 enables the agent to initially target the lower level fires, extinguishing a greater number of them before reaching the steps where the fire

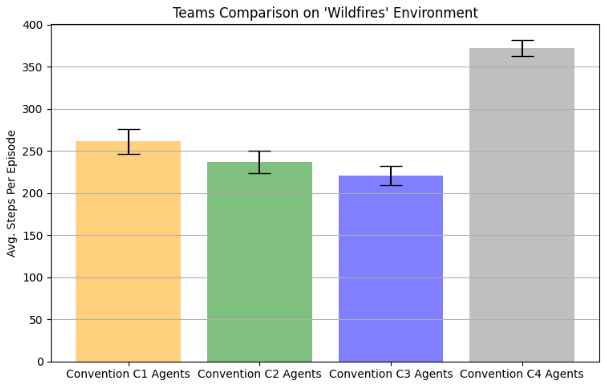

Figure 8: Results for times=100, fires=5, agents=2, seed=None, steps_incr=50

level is incremented. So, as expected, we can verify through Figure 8 that C3 has a slight advantage when the flag *--steps_incr* is active.

Impact of initialization: In our implementation, the agents' initialization defaults to a descending order of level. As a result, all Social Convention agents share the same agent convention, with the only variation occurring in the fire convention. Figure 9 illustrates that C2 exhibits a very similar performance to C3, but significantly different from C1. This is due to the fire initialization, as it was very close to an ascending order of level, being very similar to the order used in C3 but the opposite of the order used in C1. On the other hand, there can also be cases where the performance of all 3 agents is similar, in which case the fires' levels do not present a significant variance.

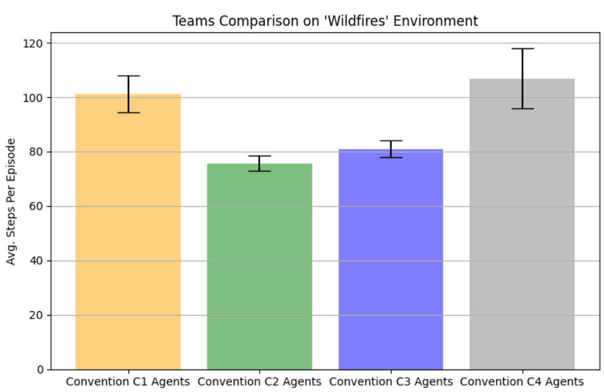

Figure 9: Results for times=100, fires=5, agents=2, seed=None, steps_incr=50

3.7 Role Based Agents

There are several noteworthy points to consider regarding the Role Based agents:

Fire Agent ratio: In scenarios where the number of agents significantly exceeds the number of fires, effective coordination amongst agents plays a crucial role in improving overall performance. By appropriately distributing their efforts amongst the fires, the agents can achieve optimal outcomes for aggravating fires (flag *--steps_incr*), as shown by the few number of total steps in Figures 10 and 11.

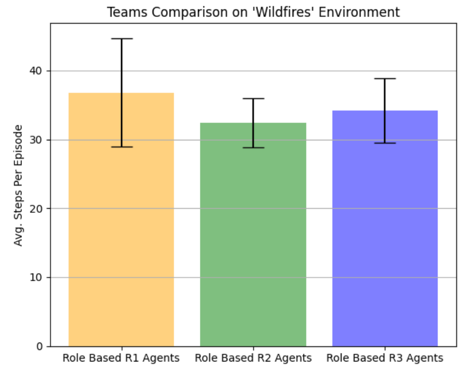

Figure 10: Results for times=10, fires=3, agents=7, seed=None, steps_incr=50

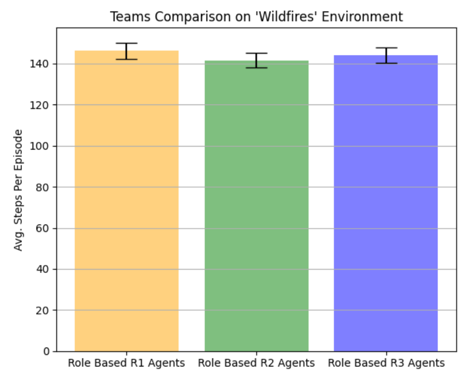

Figure 11: Results for times=1000, fires=3, agents=2, seed=None, steps_incr=50

Consequently, our role-based implementations exhibit **scalability** in accommodating larger agent populations. However, as the ratio of fires per agents approaches 1 (where the number of fires is equal to the number of agents), it is anticipated that the influence of coordination on performance will **enhance**, although not to the same degree as when the ratio exceeds one. This phenomenon occurs because, in such instances, although it mitigates the scenario where all agents attempt to extinguish the same fire tile, the behavior of these agents resembles that of heuristic agents to a greater extent, thereby restricting the potential for substantial performance enhancements. The scalability of the system can be attributed to the following factor: when there are sufficient water resources available to extinguish a fire, the remaining agents are allocated to the fires that still require resources, with priority given to those in greater need, accelerating the overall process.

Water Considerations: Referring back to the subsection on Role-based Coordination, it is noteworthy that the potential functions primarily differ in their consideration of available water. Nevertheless, as illustrated by figures 10 and 11, the performances exhibit a similar trend. This observation suggests that the role of water in these potential functions has a relatively minor impact.

Increasing Fire Intensity: Role-based coordination exhibited good performance overall, particularly when the fire intensity increases after a certain number of time steps, referred to as *steps_incr*. This characteristic posed a significant challenge for *Random*, *Pseudo Random*, and *Greedy* agents. It is worth noting, though, that the effectiveness of role-based coordination is contingent upon selecting appropriate values for *steps_incr*. Extremely low values may result in the environment becoming stuck, hindering the agents from achieving their ultimate goal.

Maximum Number of Steps: While it is true that in the tested scenarios, the performance of these agents was not affected by the maximum number of steps, as they consistently reached their goal within the allotted time, it is still important to consider the possibility where the defined maximum number of steps may not be sufficient to extinguish all fires. In such cases, it is reasonable to expect that role-based coordination would not yield optimal performance due to the limited time frame.

Role of Seed in Results Consistency: Similar to our approach with the *Greedy Agent* implementation, we conducted tests on these agents using fixed seeds to better understand how the initial conditions would influence performance. Through our analysis, we reached the conclusion that the initial seed has no significant impact on performance. This is attributed to the effective heuristics employed, which consistently deliver comparable results regardless of the specific initial conditions.

3.8 All together

Ultimately, we opted to conduct a comprehensive evaluation of all the implemented agents, assigning a distinct representation to each approach within the range of available implementations, namely those who provided better results. This process enabled us to draw conclusions regarding the most effective strategies. Therefore, the Random agents were represented by *Pseudo Random Agents*, the Greedy agents were represented by the implementation of the closest fire heuristic (referred to as *H1*), the Social Conventions agents were represented by the implementation in which conventions were defined based on the arrangement of agents in descending order of level and fires in ascending order of level (referred as *C3*) and the Role-Based agents were represented by the implementation of the distance to the closest fire tile potential function (referred as *R2*).

As anticipated, coordination demonstrates superior performance compared to random, as the coordinated approach allows agents to select actions that lead to a Nash equilibrium, taking into account the actions of other agents. The performance of random agents was significantly worse compared to the other agents. Therefore, we chose to evaluate the remaining three agents separately. The performance of random agents was significantly worse compared to the other agents. Therefore, we chose to evaluate the remaining three agents separately.

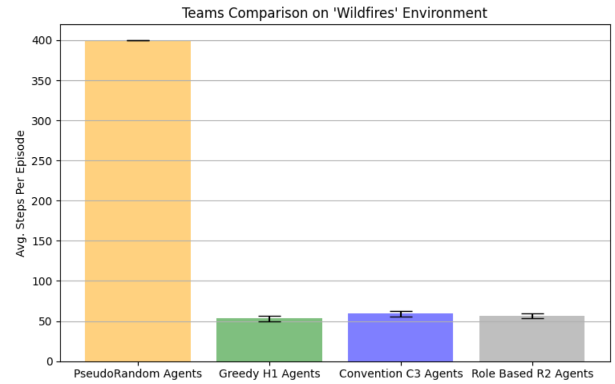

Figure 12: Results for times=100, fires=3, agents=4, seed=None, steps_incr=None

Surprisingly, the Greedy Agent performance obtained great overall results for different parameters, as seen in Figure 13. It is noteworthy that the *H1* agents do not significantly lag behind coordination strategies. In fact, in certain scenarios, the *H1* agents even manage to outperform coordination strategies by a slight margin. This result may be due to the fact the agents are more **reactive** to the environment. In the Role-based approach, the primary focus of the potential function is on the distance between the agents and the fires, and roles are assigned accordingly. However, the reassignment of roles may result in wasteful actions being taken, since the current role/target might change.

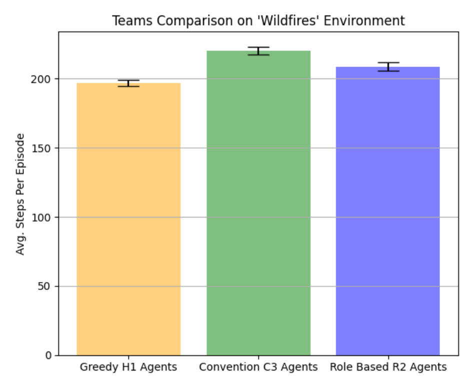

Figure 13: Results for times=1000, fires=10, agents=3, seed=103537, steps_incr=None

As the number of agents increases, there are instances where **coordination** demonstrates a **positive impact**, as intelligent decisions have a more significant influence on performance outcomes. Figure 14 illustrates that with a larger number of agents, coordination achieves slightly improved results, as expected.

As a final point, it is crucial to emphasize the significance of the initial state when comparing different types of agents. The initial state plays a vital role in shaping the performance outcomes of the agents. It is plausible that the reason why greedy agents can achieve similar performance to the employed coordination strategies lies in the specific configuration of the initial state. In another words, some initial configurations might align perfectly with the heuristics of the greedy agents, thus being the best course of action.

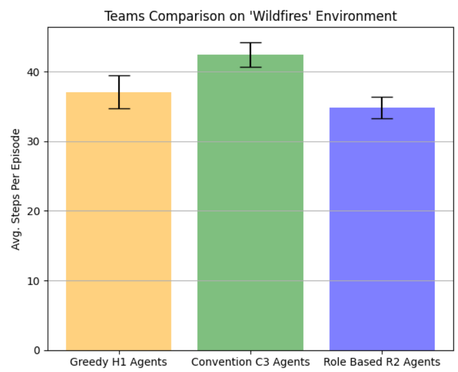

Figure 14: Results for times=200, fires=5, agents=10, seed=None, steps_incr=None

4 CONCLUSION

We set out to find the best possible strategies to coordinate efforts towards dealing with wildfires while efficiently managing resource allocation. For that reason, we successfully implemented a multi-agent system that simulates those scenarios and concluded that coordinated efforts can have a great performance (as demonstrated by our coordination agents), although we couldn't completely discard the some of the greedy approaches that we explored.

Not all possible agents were implemented, mostly due to time constrains, and as such are left bellow as future work:

- **Rational Agents:** In this project, the incorporation of rational agents utilizing algorithms such as Q-learning could prove advantageous. By leveraging Q-learning, these agents would have the ability to learn from past coordination attempts, enabling them to enhance their coordination capabilities over time. This learning mechanism would facilitate improved coordination amongst the agents, leading to more effective outcomes in future coordination tasks.
- **Having Trucks and/or Helicopters with different types of agents:** During the development phase, our focus primarily revolved around treating trucks and helicopters as identical agents. However, this approach opens up possibilities for incorporating multiple types of agents simultaneously and exploring how their interactions could mutually influence one another. By considering diverse agent types, we can delve into the dynamics and potential interplay between different agent categories, thereby gaining valuable insights into their collective impact on the overall system
- **Limited sight:** Sight plays a crucial role in both greedy and coordination based agents. Our focus was mostly on allowing the agents to see the full grid (i.e, they always knew where the fires and water sources were located), simulating a real life scenario. Having agents with limited sight might open up more heuristics and might impact performance of the already implemented agents. For example, since they would not be aware of all fires, they might not be able to correctly identify the strongest fire, in the case of the strongest fire heuristic for instance.

All in all, this was a successful project and we hope to have made contributions to resource management and allocation strategies, when dealing with wildfires.

REFERENCES

[1] LBF. 2023. LBF. (May 2023). Retrieved May 11, 2023 from https://github.com/semitable/lb-foraging
[2] Matplotlib. 2023. Matplotlib. (May 2023). Retrieved May 10, 2023 from https://matplotlib.org/
[3] NumPy. 2023. NumPy. (May 2023). Retrieved May 10, 2023 from https://numpy.org/
[4] Python. 2023. Python. (May 2023). Retrieved May 9, 2023 from https://www.python.org/
[5] scikit learn. 2023. scikit-learn. (May 2023). Retrieved May 10, 2023 from https://scikit-learn.org/