Neel Jhangiani, 1975593

Hyundoo Jeong, 2212332

Jurnae Jones, 1957831

Mitchell Jackson, 1607912

Path Discovery in a 3-Agent Transportation World

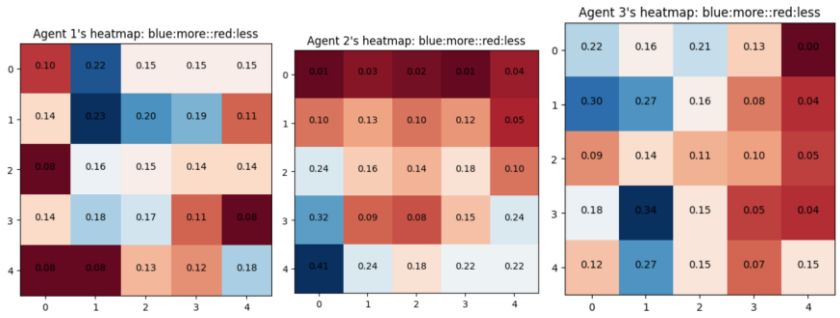
Using Reinforcement Learning

Introduction

In this project we intend to explore a 3-agent reinforcement learning scenario where agents collaborate to move blocks from pickup locations to drop off spots on a 5x5 grid world. The agents, red, blue, and black operate in sequence and can only move North, South, East, or West. However, two agents cannot occupy the same cell simultaneously, which is a crucial constraint. We will utilize three policies; PRandom, PGreedy, and PExploit in our experimentation. In this experimentation, we intend to compare the performance between the traditional Q-learning algorithm and the SARSA algorithm, explore how the learning rate affects the performance, as well as investigate the impact of changing pickup locations on the agent's performance.

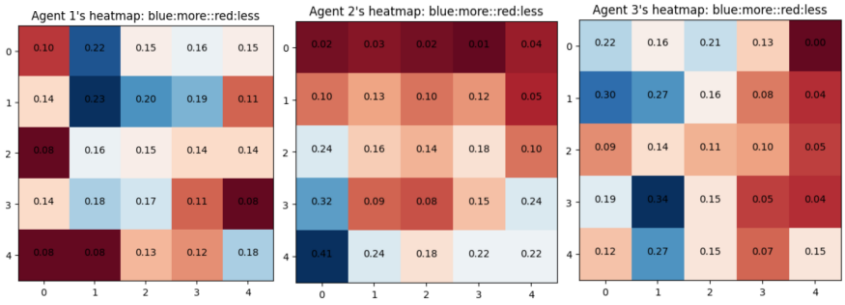
Experiment 1

1. Start of random policy (500 steps)

a. 

b. Our first 5 steps implement a policy that chooses an action at random if the pickup or dropoff actions are not applicable. We can, however, see how this policy is focusing more on exploration than exploitation. The “frequently visited” grid cells are not related to the environment's attributes or how the strategy can accumulate awards. Even though the actions taken with regards to this policy may not be optimal, this could still be beneficial when it comes to information gathering and understanding.

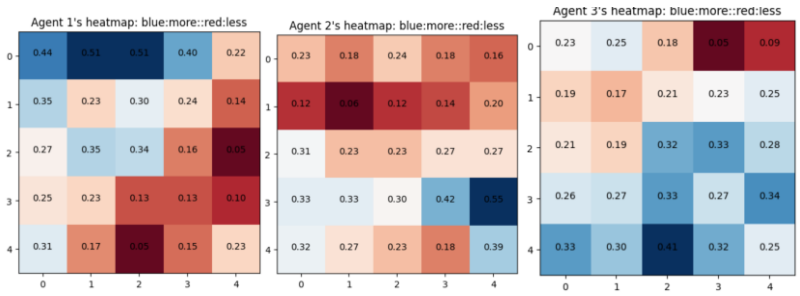
2. Continuation of random policy

a. 

b. As we continue the random policy for 8500 more steps, we see that there is very little change to the frequency maps for each agent. The agents are not learning anything from their actions at this point in the experiment. This was expected because this policy doesn’t

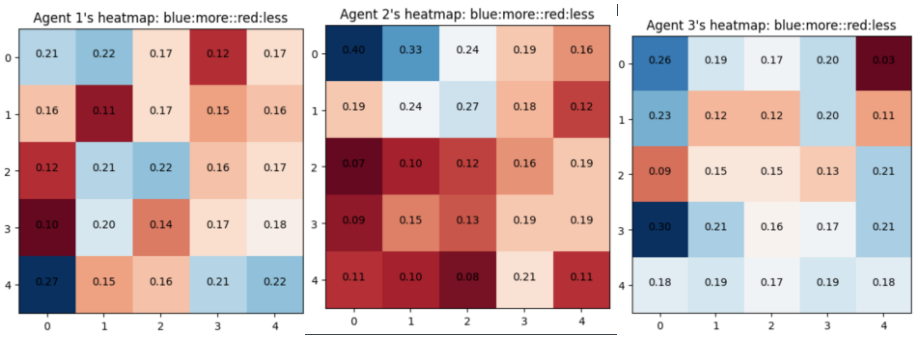
exactly prioritize learning or applying its knowledge for a certain benefit (like reward accumulation).

3. Greedy policy

a. 

b. In this part of the experiment where we implement the greedy policy, we can actually see the agents learning and applying their knowledge. We can see that the agents are avoiding collisions with each other. Each agent seems to have a general area on the grid where they can execute their actions. By prioritizing actions that yield the highest reward, this policy introduces more exploitation compared to the first 500 steps (of the random policy).

4. Exploit policy

a. 

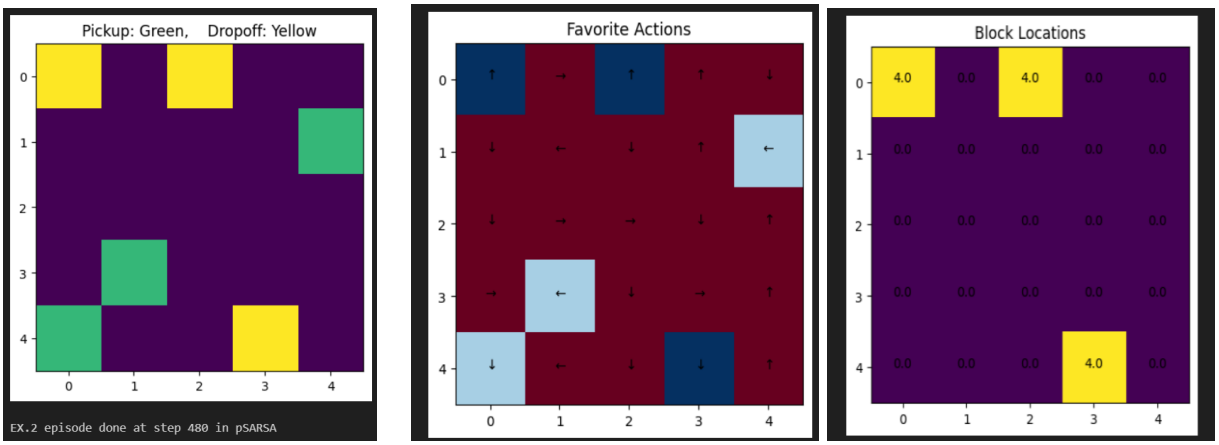
b. Similar to the greedy policy, we can see the agents applying their knowledge to take more optimal actions. The heat map shows slightly more subtle color in some places and more intensity in

others. I think this is very indicative of its balance of exploration v. exploitation. The epsilon value of 0.2 adds randomness to the algorithm and therefore leads to more exploration. That's why we can see more confidence (intensity) the agent's actions. Meanwhile the use of the q-table actually applies that knowledge and uses it for some benefit which is why we see more “intense” spots near areas with higher reward.

Experiment 2

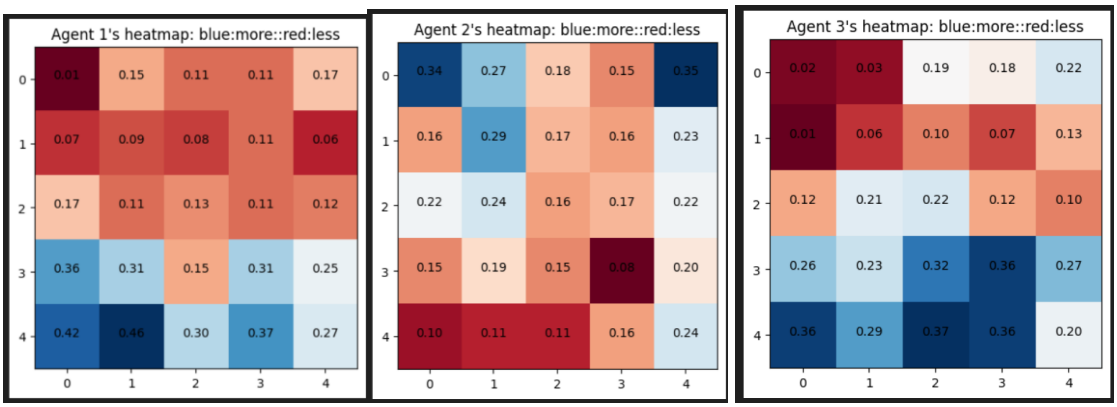
In experiment 2, we constructed a SARSA algorithm to compare the performance with Q-learning algorithms in a three-agent reinforcement learning environment. In our project, we implement our SARSA algorithm that each agent Q-table was initialized with zero values, representing their lack of knowledge about the environment. We adopted an epsilon-greedy policy(pxploit) to balance exploration with exploitation. This policy allowed agents to select random actions, facilitating exploration of the state-action space. Then, the agents updated their Q value based on a combination of reward. This update happened inline with the SARSA algorithm on-policy, where in the current action and the next action are both derived from the same policy. Our algorithm incorporated a mechanism to ensure that no two agents could occupy the same space at the same time, effectively minimizing blockages and enhancing coordination.

After conducting 9000 steps for SARSA, we got those results in terms of demonstrating block transportation tasks.



Our test successfully completed the task within 480 steps above the first left image. The “Favorite Actions” visualization indicated that the agents learned to prioritize picking up and dropping off actions. For example, upper grid movement which shows leftward and rightward arrow, suggesting the most consistent actions by agents. On the other hand, the red and light blue grid represented less consistent actions. That is, Agents can learn behavior by looking at dark blue grid and light blue grid. It is also helping to build strategy for what is the most effective movement.

The “block location” demonstrates the final state of the PD-world grid at the conclusion of SARSA algorithm episode. The “4.0” confirms that agents have successfully transported blocks to these designated dropoff points, fulfilling the objective of the simulation. Other “0.0” implies that agents did not leave any block at non-drop off location, suggesting efficient retrieval and delivery strategies. This distribution of blocks shows the successful completion of the task by agents, indicating effective learning and problem-solving capabilities as instilled by the SARSA algorithm.

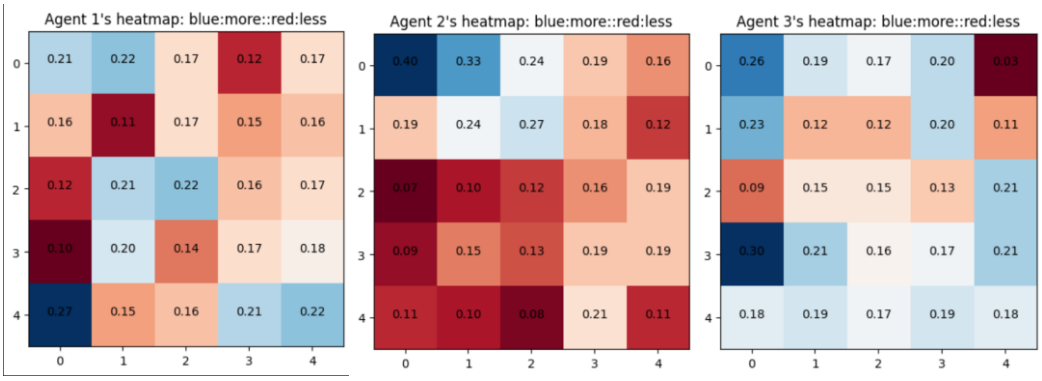


The heatmaps of all three agents reveal distinct patterns of movements across the PD-world. Agent 1 shows a preference grid lower boundary, Agent 2 for the grid's perimeter, and Agent 3 for the grid lower center part. The varied patterns suggest that the agents may have divided the grid among themselves to minimize overlap in their routes, which would be an efficient strategy to maximize coverage and minimize blockages in a multi-way. The presence of red areas across the heatmaps for each agent further indicates that certain paths are frequently traveled, the agents retain a level of exploratory behavior in less frequented areas, likely in response to the changing state of the environment as blocks are picked up and dropped off.

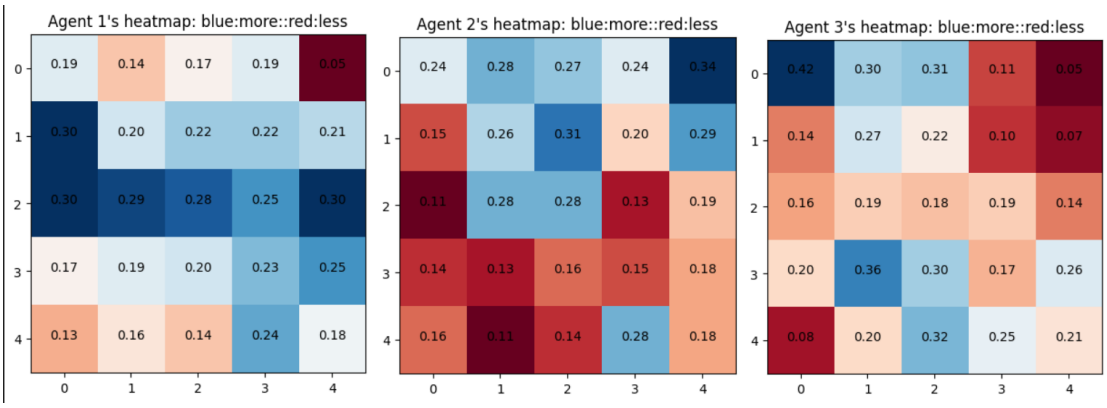
Experiment 3

A significant parameter in reinforcement learning techniques, such as Q-learning, is the learning rate (alpha). This rate governs how new data impacts existing information when revising Q-values. In this experiment, we are evaluating how changing the learning rate affects the performance of the Q-learning algorithm the agents take. We conducted the experiments over 9000 steps, with each learning rate being tested

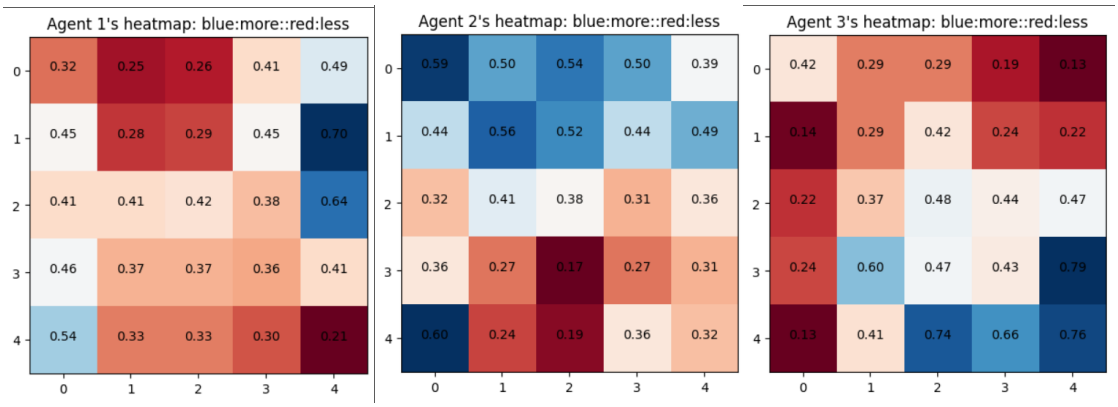
individually. The first 500 steps utilized the PRandom policy, and we finished the remaining 8500 steps using the PExploit policy. We altered the gamma parameter to be 0.9 when experimenting with the learning rates as it gave us better results. The results are as follows; with a learning rate 0.3, the algorithm finished in 430 steps. With a learning rate of 0.15, it finished in 20 steps. Lastly, with a learning rate of 0.45, it finished in 484 steps. The learning rate of 0.15 led to a much faster convergence than the others, with the agents completing the tasks in only 20 steps. However, this fast convergence could indicate a problem about the quality of the learned policies. This could be due to the lower learning rate hindering their ability to adapt to changes in the environment correctly, overlooking information or making premature decisions. When the learning rate is 0.45, agents update their Q-values more aggressively based on new experiences, leading to the slowest convergence out of the three. A higher learning rate could cause instability in the learning process, causing the collaboration of the agents to be suboptimal. The learning rate of 0.3 provides a balance between exploration and exploitation, allowing the agents to learn efficiently while avoiding excessive nuances. This moderate learning rate seems to be well suited, completing the task in 430 steps. We will now visualize the differences between the different learning rates.



The figures above show the frequency an agent moves to a certain spot with a learning rate of 0.3. We can see some sort of understanding between the agents as they are avoiding each other.



The figures above are the agents with a learning rate of 0.15. We can clearly see that the agents are not coordinating well with each other as they have the same frequency as each other. The algorithm was able to converge the fastest, but raises a problem in the quality.

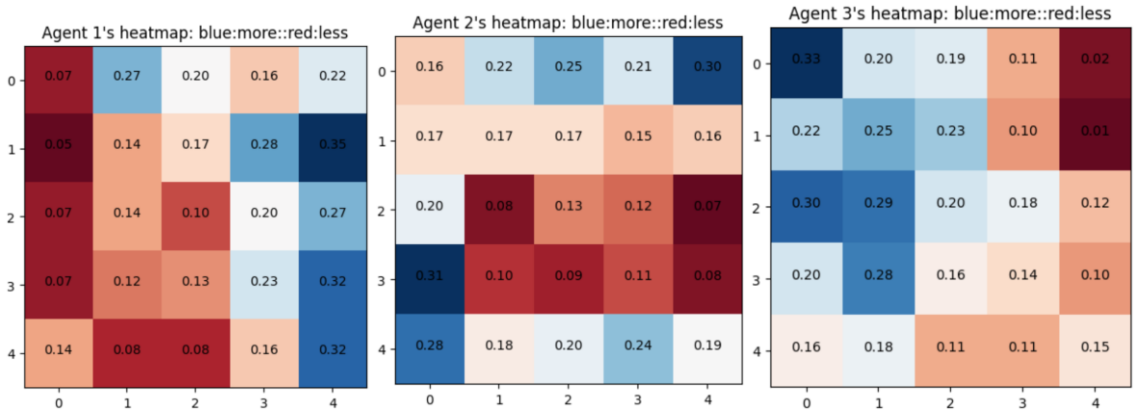


The figures above are the agents with a learning rate of 0.45. We can see that the agents are working really well together, avoiding each over when needed. However, this led to an overall slower convergence.

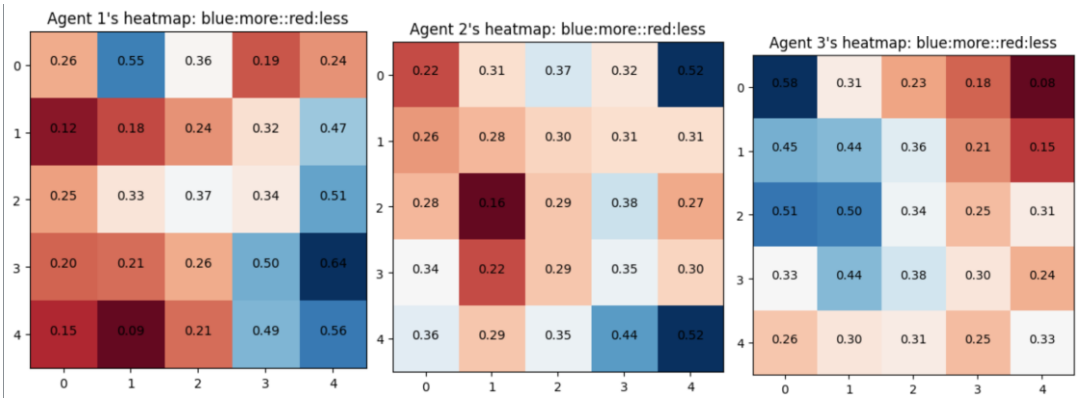
The learning rate is an essential factor that influences how well the agents performed in this 3-agent reinforcement learning task. This experiment revealed that a moderate learning rate of 0.3 yielded better results compared to lower or higher rates. A low learning rate makes it difficult for agents to adapt and learn effectively, while a high rate can cause instability and erratic behavior.

Experiment 4

In experiment 4 we got a closer look at how well our agents can forget or unlearn old behaviors and learn and adapt to new changes. We have established that together, our strategy and the exploit policy avoids collisions and maximizes the accumulation of reward values. The first part of experiment 4 implements experiment 1c again, but continues until the algorithm has reached the termination state for the third time. When we compare our first implementation of the exploitation policy in experiment 1c and this implementation, we see a lot of consistencies. We can tell that the model continuously learns through more iterations. The colors on the heat map for experiment 4 are more vibrant or intense which suggests a higher level of confidence that the model has now that it has learned from 3 complete episodes. Below are the frequency maps for each agent in part 1 of Experiment 4:

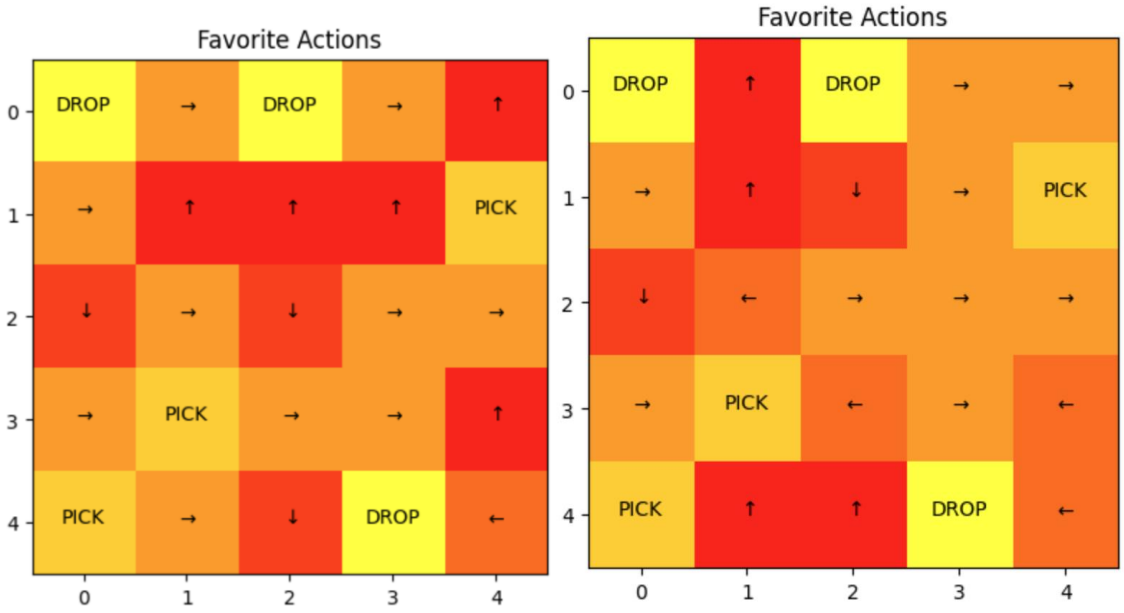


The next part of experiment 4 changes the location of the pickup spots, and continues the exploitation policy. The agents have to unlearn the behaviors associated with exploiting the old environment and learn new behaviors that exploit the new environment with new pickup spots:



The change in the heat maps indicates how certain some frequent grid cells become less visited as the agent adapts to the new environment. 2 of the old pickup locations were at the top right of the grid and we can see that the Agents are still in the process of unlearning the behaviors associated with exploiting the old environment. However, there are patterns shown on the heat maps that suggest that the agents are slowly starting to learn and exploit the new pickup spots. In general, there is also less confidence in each

agent's decision due to the amount of uncertainty and the sudden change in the environment. Overall, the agent heat maps that represent the experiment with the new pickup spots are very indicative of the learning and unlearning process. The slight scattering of the heat map suggests that the agents' behaviors are in the process of being changed to exploit the new pickup spots.



The graphs above show each cell's favorite action to take (left for experiment with old pickup locations; right for experiment with new pickup locations). These graphs are also indicative of the progress that the agents are making to unlearn the old environment. While many of the actions that were taken to exploit the old environment have been unlearned, the algorithm will probably need more opportunities to explore and learn about the new environment to fully utilize its exploitation policy.

Conclusion

Overall, our analysis reveals that the traditional Q-learning algorithm performs slightly better comparing it to the SARSA algorithm. This is because SARSA can use an exploration step in the second step, which causes it to converge to a solution slower. However, when comparing the heatmaps for both algorithms, it can be seen that the agents follow a certain path better using the SARSA algorithm than the traditional Q-learning algorithm. When comparing the performance of the different learning rates, our analysis shows that a moderate learning rate of 0.3 offers a balanced approach between exploration and exploitation. Although lower and higher learning rates lead to faster and slower convergence rates, respectively, the quality of the learnt policies was reduced. Lastly, when exploring how well the agents adapted to change, we can argue that the agents had some trouble unlearning their old environment. This could be due to the fact that they exploited that environment to their fullest. However, over time, they were able to start to adapt to the new environment. If we allow the agents to continue for more steps, it can be said that the new environment will be fully learnt by the agents.