Comparison of SARSA and Q-Learning Algorithms in a Deterministic Grid World Environment

Presented to professor: **Sidney Givigi**

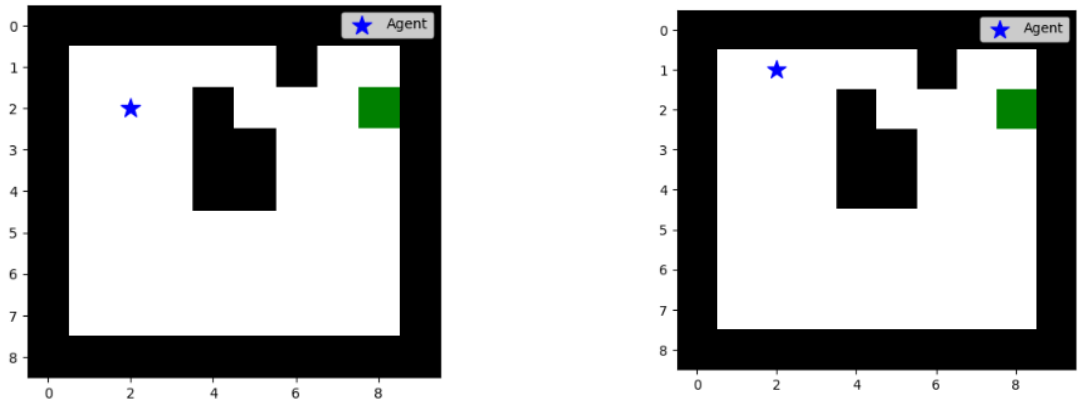
By: **Amr Mostafa Ibrahim**

I. Introduction

This report delves into the comparative analysis of Q-learning and SARSA algorithms within a deterministic grid world environment. The adjustments made to both implementations were motivated by my curiosity to explore how each algorithm approaches and solves identical tasks, aiming to uncover their distinct methods in reinforcement learning. Visualizations depicting the agent's trajectory from its starting point to the goal provide visual insights into their decision-making processes and adaptive strategies.

The decision to modify both implementations stemmed from a keen interest in understanding how Q-learning and SARSA algorithms navigate the same grid world environment. This exploration required significant time and effort to accurately capture and visualize how each algorithm adapts its strategies under different parameter settings. By comparing their performance metrics, such as average rewards and steps taken, I aimed to discern which algorithm achieves optimal solutions effectively.

The environment consists of a 9x10 grid featuring predefined start and goal positions, alongside various blocked cells, mirroring the setup detailed in the uploaded notebook on OnQ. The agent's objective is to navigate from the start to the goal, maneuvering around blocked cells to minimize steps and maximize average rewards.



(A) (B)

Fig.1 The grid world environment. **A**) Initial state for agent (2, 2). **B**) agent reached state (1, 2), received reward (0) and it will receive -5 in case of blocked cell and 10 if it reaches to goal at (2,8).

The remainder of this report is structured as follows: **Section II** discussed algorithms that we implemented Section **III** illustrate criteria that we used in our comparison between different algorithms **Section IV** mention results and discussed it with deterministic environment and show the effect of changing parameters such as (**epsilon and alpha)** Finally, **Section V** concludes the report with a discussion on findings.

II. Algorithms

SARSA Algorithm (State-Action-Reward-State-Action)

It is a method within reinforcement learning where an agent learns by interacting with its environment. The agent operates based on a predefined strategy, receives rewards, and refines its knowledge to enhance future decision-making. SARSA is categorized as an "on-policy" approach because it updates its understanding using actions taken according to its current strategy. The process starts with initializing Q-values, which represent the merit of taking specific actions in given states and adjusts these values as the agent explores its surroundings. At each step, the agent observes its current state, selects an action, receives a reward, observes the subsequent state and the action it would take there, and updates the Q-value based on these observations. This update rule considers the immediate reward and the anticipated value of the next action, ensuring that the learning process aligns with the ongoing strategy.

Q-learning Algorithm

It's another approach within reinforcement learning, distinct from SARSA as an "off-policy" algorithm. This means it learns the best policy's value independently of the current actions. Q-learning begins by setting up Q-values and adjusting them based on the agent's interactions with the environment. At each step, the agent selects an action, receives a reward, and observes the subsequent state. The Q-value is then updated using this reward and the highest Q-value of the following state, regardless of the next action under the current policy. This method allows Q-learning to converge faster to the optimal actions by consistently considering the potential for the highest future rewards, not just those influenced by the current policy.

III. Evaluation Criteria

To effectively compare the performance and efficiency of SARSA and Q-learning algorithms, we employ several evaluation criteria. These criteria provide a comprehensive assessment of each algorithm's strengths and weaknesses, helping us determine their overall effectiveness and practical applicability. The key evaluation metrics are as follows:

Learning Time: This criterion measures the duration taken by the algorithm to converge to an optimal policy. It indicates the efficiency of the learning process and how quickly the agent can learn the best actions to take in the environment.

Steps: This refers to the total number of steps the agent takes to reach the goal from the starting position. It serves as an indicator of the agent's path efficiency; fewer steps suggest a more optimal path.

Average Reward: This criterion calculates the mean reward accumulated by the agent over multiple episodes. It provides insight into the overall performance and effectiveness of the agent's policy, with higher average rewards indicating better performance.

IV. Results and Discussion

1. Comparison in Deterministic Environment (Noise =0)

1.1 alpha=**0.1**

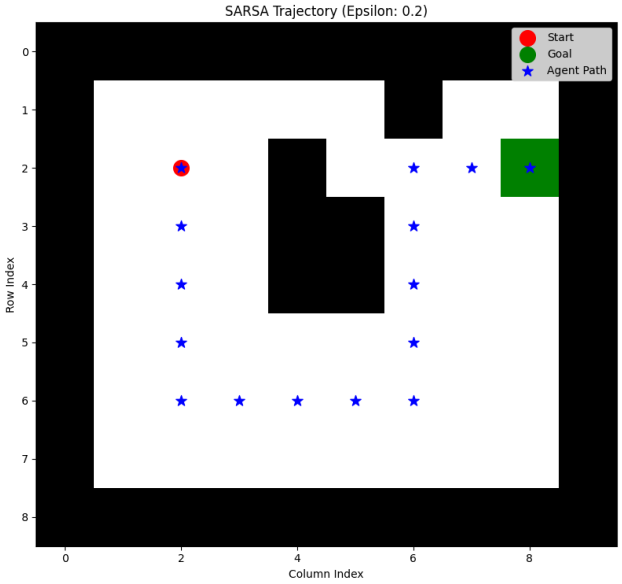
- Epsilon =**0.2**

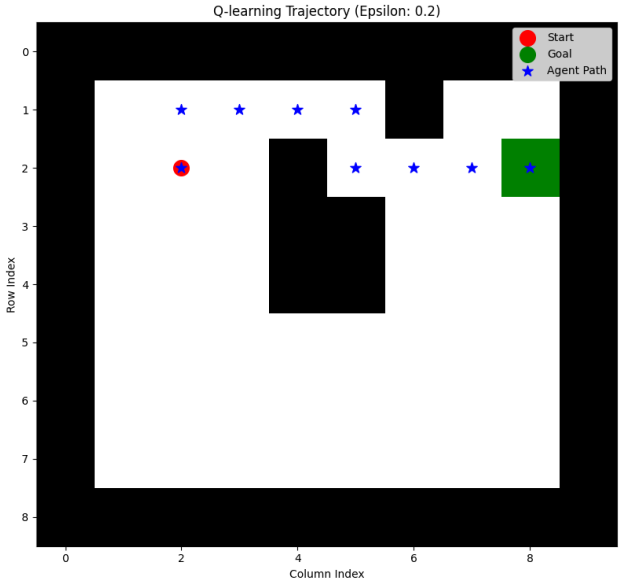
<u>SARSA</u>

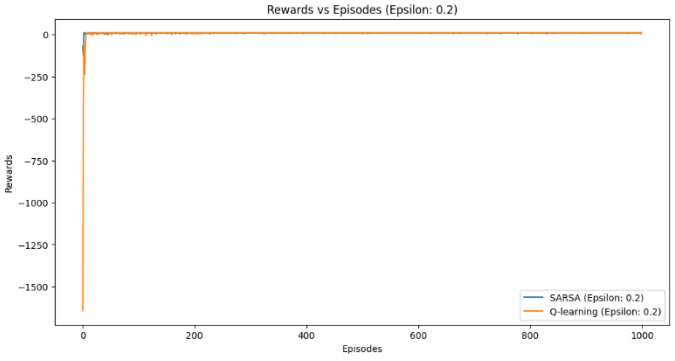
It reached an optimal policy in 0.06 seconds, demonstrating rapid convergence. The agent took 14 steps to achieve the goal, showing an efficient path, though not the shortest possible. With an average reward of 9.605, it consistently navigated towards the goal successfully, avoiding penalties.

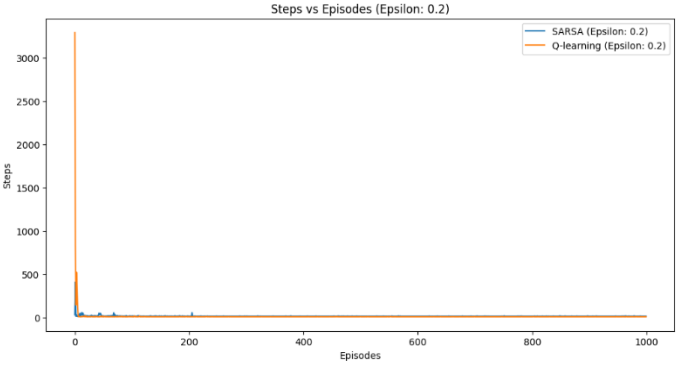
Q-learning

Q-learning completed its convergence in only 0.03 seconds, highlighting its effective learning efficiency. The agent reached the goal in just 8 steps, demonstrating a more direct and optimal path than SARSA. However, despite its speed and directness, Q-learning achieved a lower average reward of 7.09 compared to SARSA. This difference may indicate that Q-learning took more risks or encountered slightly more penalties during its path to convergence.









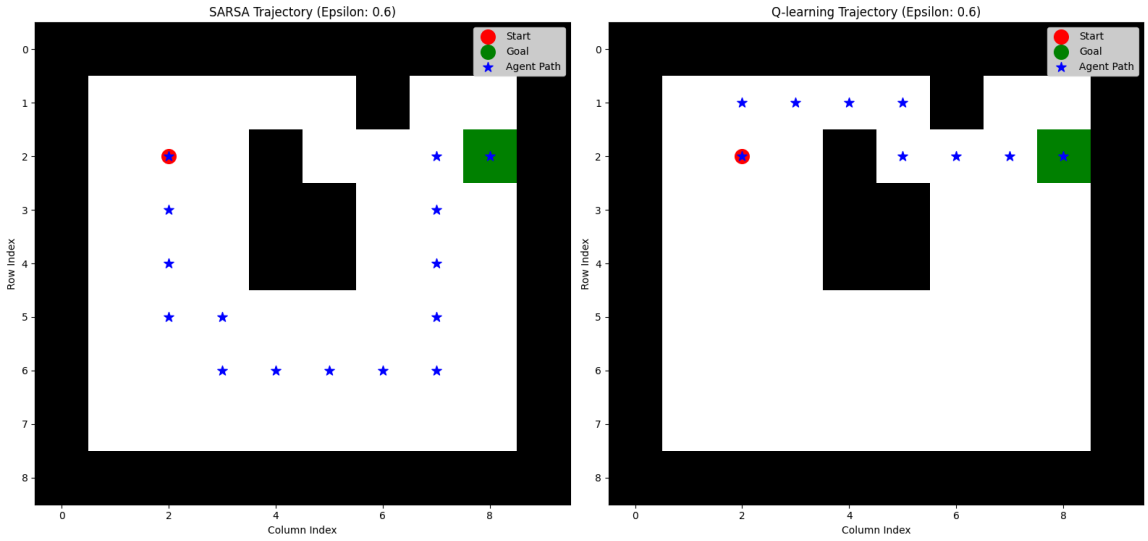
➢ Epsilon =**0.6**

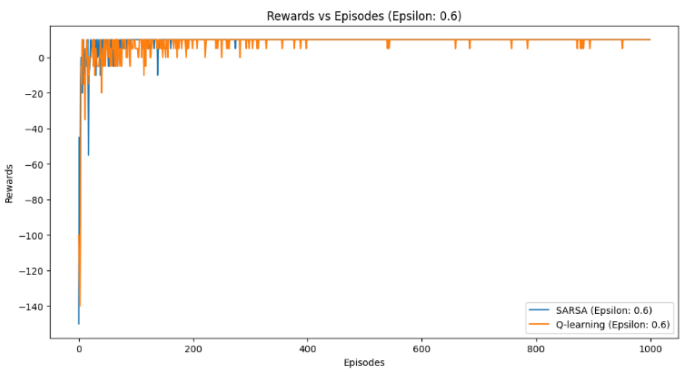
SARSA

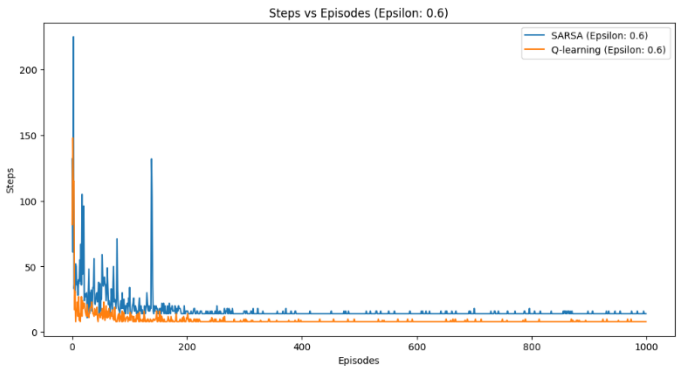
It took 0.10 seconds to reach convergence, slightly longer than with Epsilon = 0.2. The agent still needed 14 steps to achieve the goal, maintaining similar efficiency. With an average reward of 9.05, it showed consistent performance, albeit slightly lower than with Epsilon = 0.2, likely due to increased exploration.

Q-learning

Q-learning also demonstrated effectiveness with Epsilon = 0.6, achieving convergence in 0.06 seconds. The agent once more reached the goal in 8 steps, matching the performance under the previous Epsilon setting. However, the average reward of 8.3 was inferior to both SARSA with Epsilon = 0.2 and Q-learning with Epsilon = 0.2, suggesting possible exploration or riskier decision-making due to the higher exploration rate.







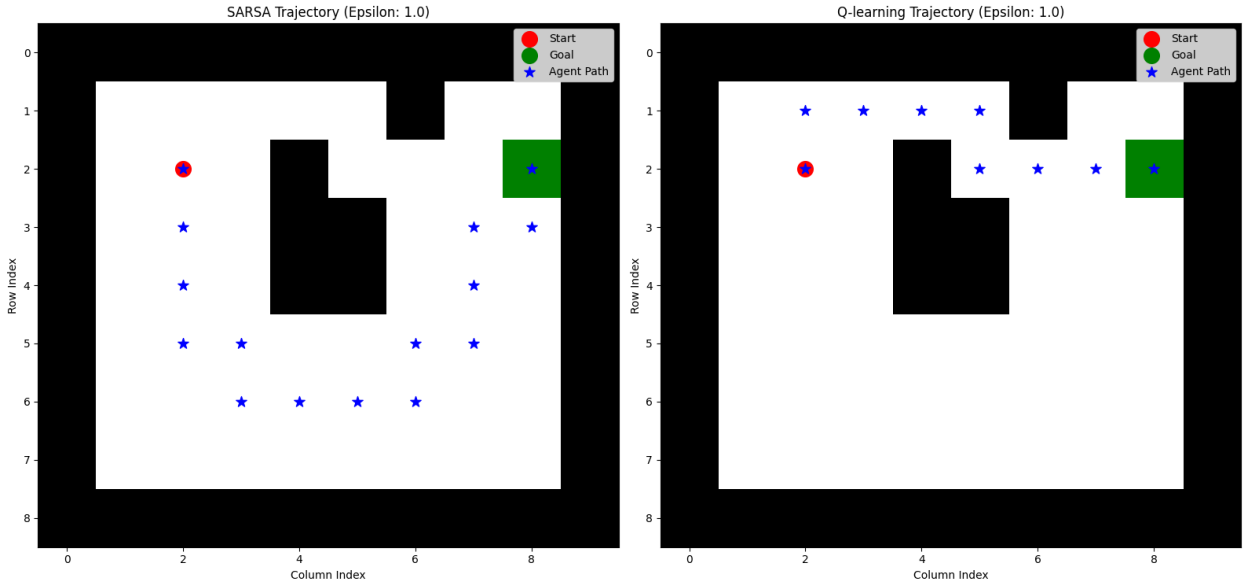
➢ Epsilon =**1**

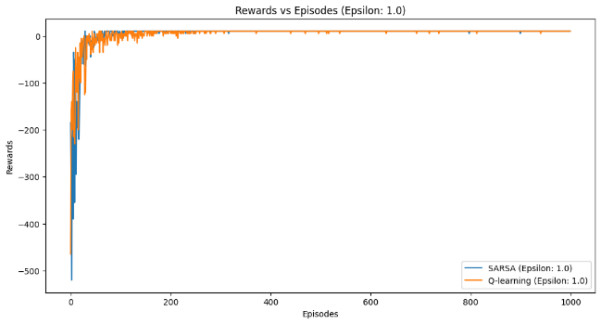
SARSA

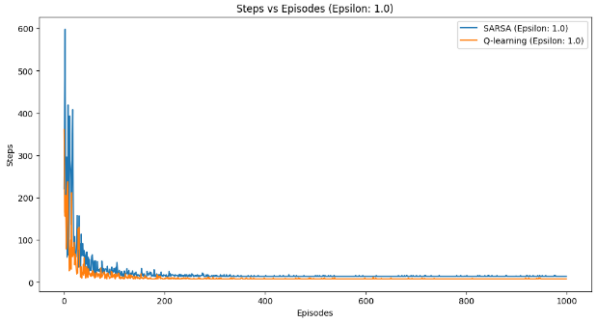
Using a high exploration rate (Epsilon = 1), achieved convergence in just 0.08 seconds, demonstrating effective learning efficiency. The agent navigated to the goal in 14 steps, consistent with past results. However, the average reward dropped to 4.48, suggesting that increased exploration resulted in more penalties or suboptimal decisions.

Q-learning

With a high exploration rate (Epsilon = 1) also demonstrated rapid convergence, achieving this in just 0.05 seconds. The agent reached the goal in 8 steps, indicating efficient pathfinding. Despite the exploration, the average reward of 4.685 slightly surpassed SARSA with Epsilon = 1, implying better decision-making or fewer significant penalties.







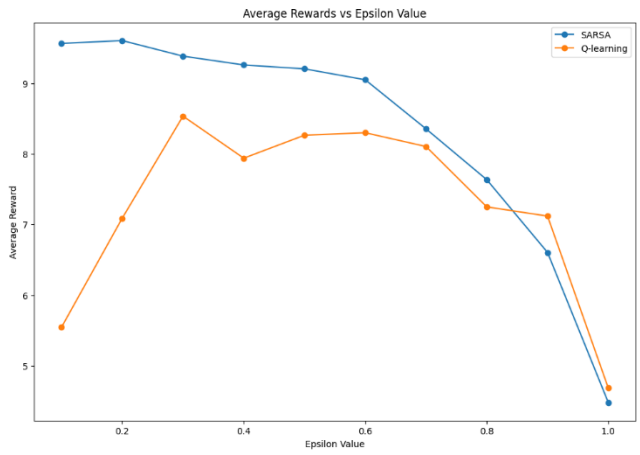
Comparison with Epsilon = 0.2, Epsilon = 0.6 and Epsilon = 1 for alpha = 0.1:

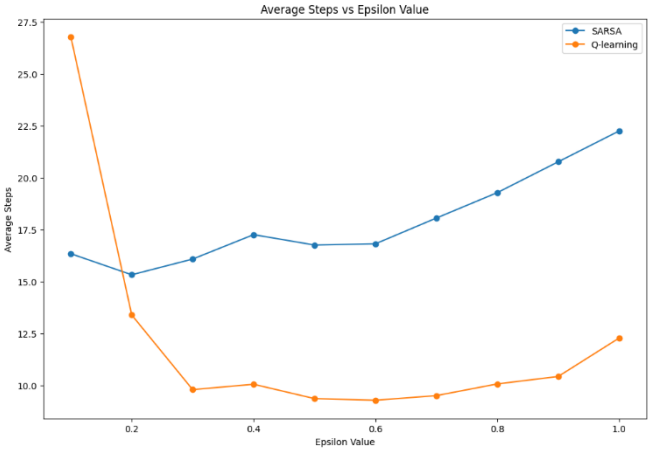
When compared to Epsilon values of 0.2 and 0.6, SARSA and Q-learning with Epsilon set to 1 demonstrated quicker learning speeds (0.08 seconds for SARSA and 0.05 seconds for Q-learning) but achieved lower average rewards (4.48 for SARSA and 4.685 for Q-learning). This suggests that despite the higher exploration rate at Epsilon = 1, which facilitated faster learning, the resulting strategies were less effective. This was evidenced by more penalties incurred or less efficient navigation towards the goal compared to lower exploration rates.

Optimal Epsilon Analysis:

Based on average rewards, SARSA performed best with Epsilon set to 0.2, achieving the highest average reward (9.605), which indicates a good balance between exploring new options and exploiting known paths. In contrast, Q-learning achieved its highest average reward (7.09) with Epsilon set to 0.3, suggesting that a slightly higher exploration rate was advantageous for this algorithm. However, when considering the number of steps taken to reach the goal, SARSA demonstrated optimal performance with Epsilon set to 0.2, indicating that less exploration was sufficient to find a direct path. On the other hand, Q-learning performed optimally with Epsilon set to 0.6, suggesting that a higher exploration rate helped in discovering a more efficient path with fewer steps.

These findings reflect the trade-off in reinforcement learning between exploring new possibilities and exploiting known strategies. Higher Epsilon values, such as 1 in this context, prioritize exploration, leading to faster learning but potentially lower average rewards due to increased penalties or less optimal paths. Lower Epsilon values strike a better balance, promoting more efficient strategies that yield higher rewards, even though they may require longer learning times.





1.2 alpha=**0.3**

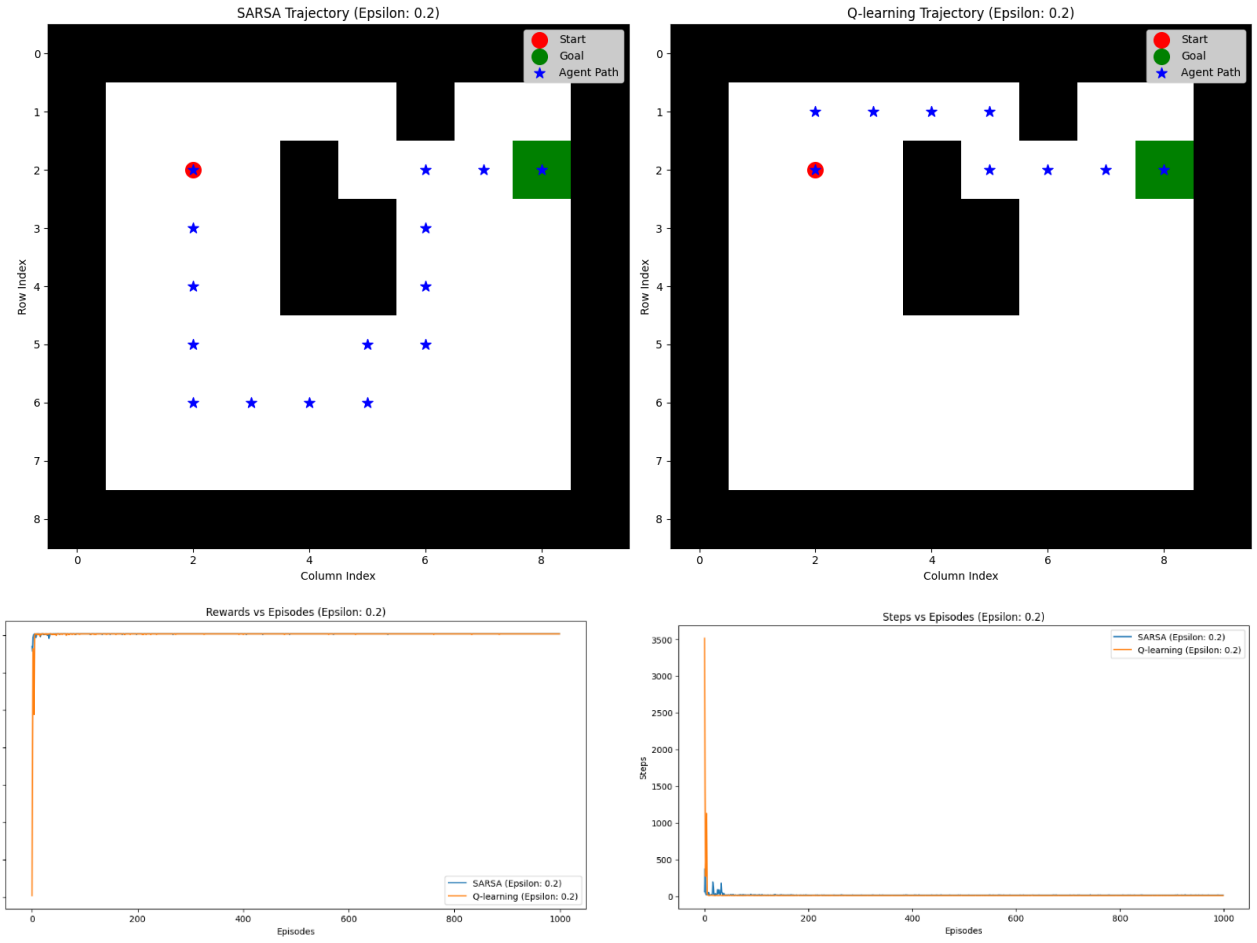
- Epsilon =**0.2**.

SARSA

It achieved convergence to an optimal policy in a speedy 0.06 seconds. The agent reached the goal in 14 steps, showing an efficient path, though not the shortest possible. With a high average reward of 9.51, SARSA demonstrated consistent performance and successful navigation to the goal without incurring penalties.

Q-learning

It achieved convergence in a mere 0.03 seconds, demonstrating its efficiency in learning. The agent navigated to the goal in just 8 steps, showing a more optimal and direct path compared to SARSA. However, despite this efficiency, Q-learning's average reward of 6.395 was lower than SARSA's, possibly indicating that it took more risks or encountered slightly more penalties in achieving its faster path.



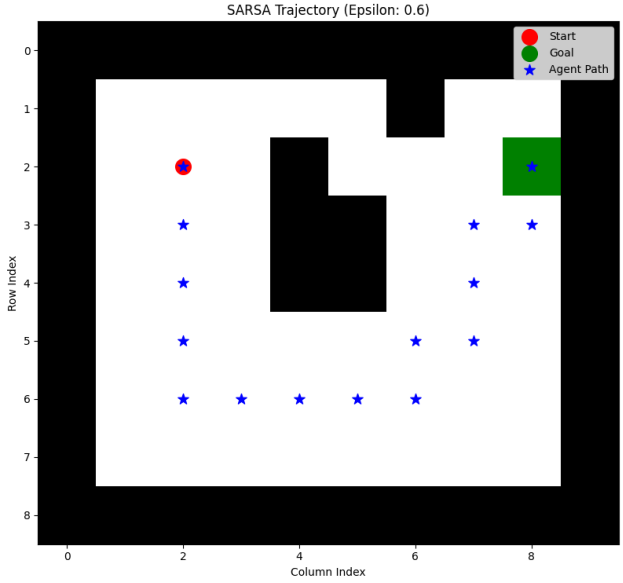
- Epsilon =**0.6**

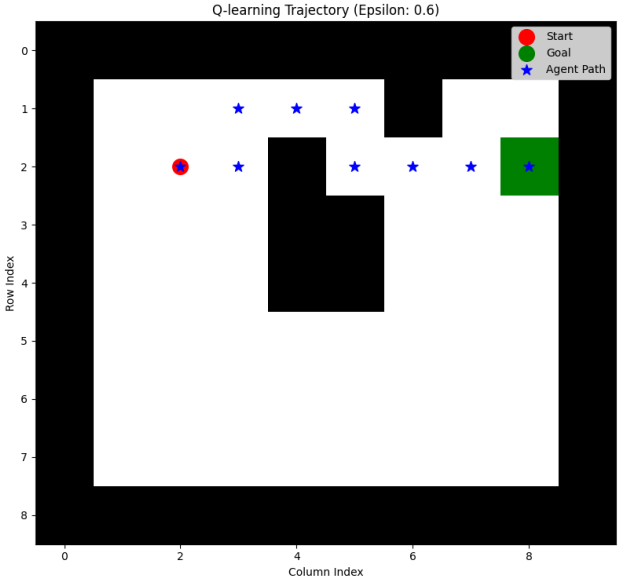
SARSA

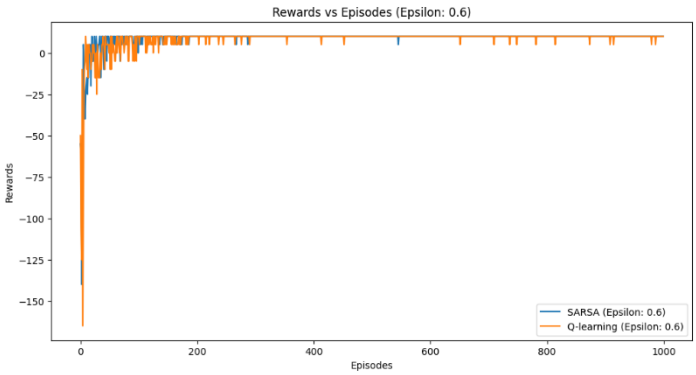
It achieved convergence in 0.05 seconds, slightly faster than with Epsilon set to 0.2. The agent once more took 14 steps to reach the goal, showing comparable efficiency. With an average reward of 8.84, SARSA demonstrated consistent performance, though slightly lower than with Epsilon set to 0.2, likely due to increased exploration.

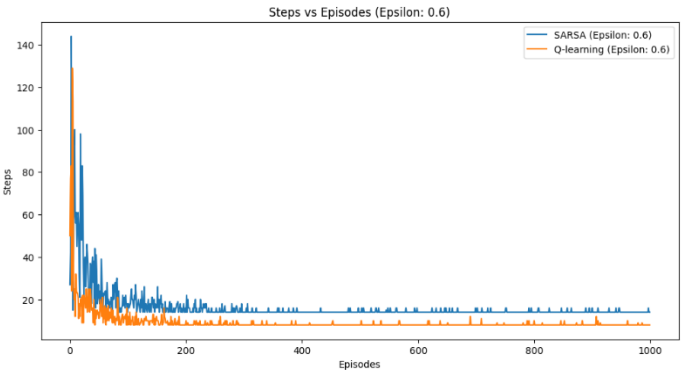
Q-learning

Q-learning demonstrated efficiency with Epsilon set to 0.6, achieving convergence in 0.03 seconds. The agent again reached the goal in 8 steps, maintaining consistency compared to the previous Epsilon setting. With an average reward of 8.27, this was higher than Q-learning with Epsilon set to 0.2, suggesting that the increased exploration rate contributed to finding a more optimal path with fewer penalties.









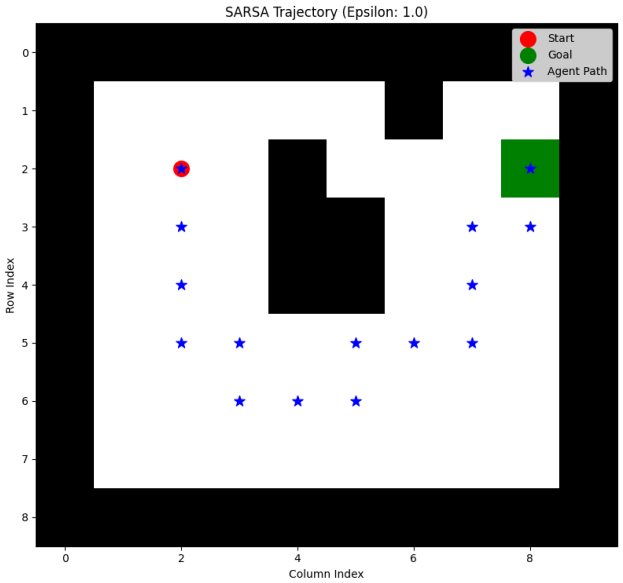
- Epsilon =**1**.

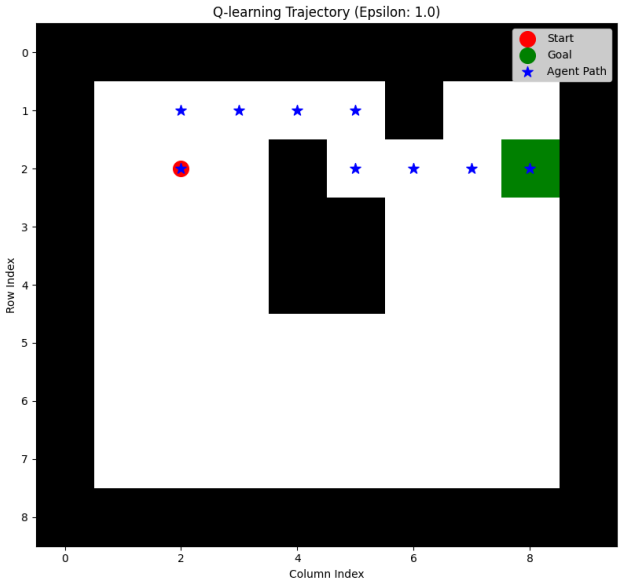
SARSA

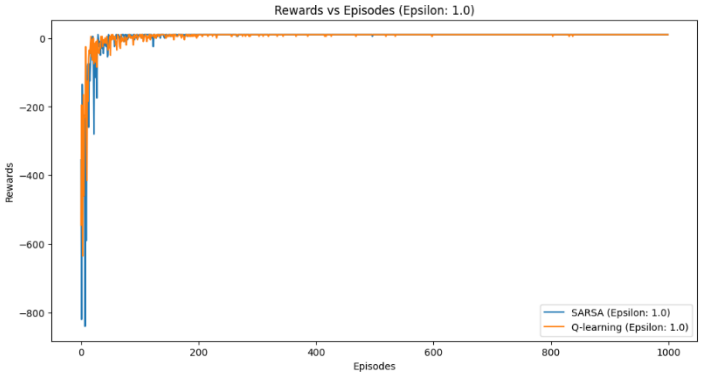
It achieved convergence in 0.07 seconds with Epsilon set to 1, demonstrating efficient learning despite increased exploration. The agent navigated to the goal in 14 steps, consistent with previous runs. However, the average reward dropped to 2.82, suggesting that the higher exploration rate resulted in more penalties or less effective decision-making.

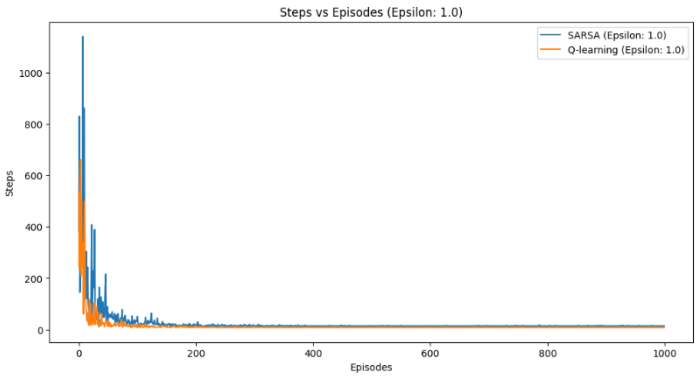
Q-learning

Q-learning converged rapidly in 0.04 seconds with Epsilon set to 1. The agent reached the goal in 8 steps, demonstrating efficient pathfinding. With an average reward of 3.52, this was higher than SARSA with Epsilon set to 1, indicating potentially better decision-making or fewer penalties despite the higher exploration rate.







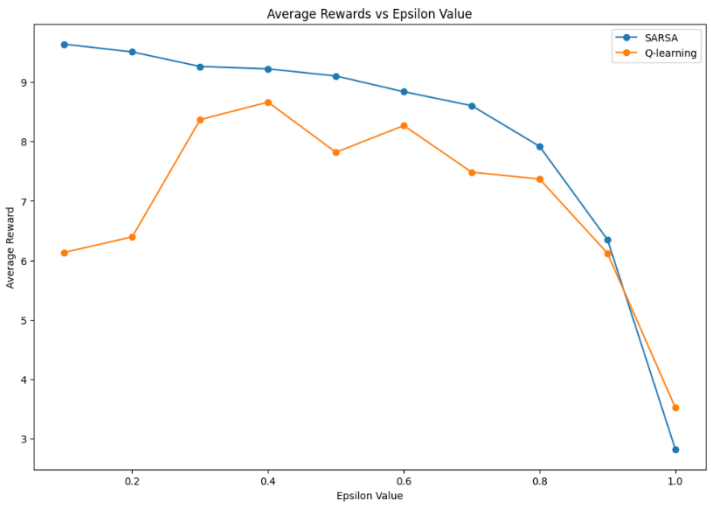


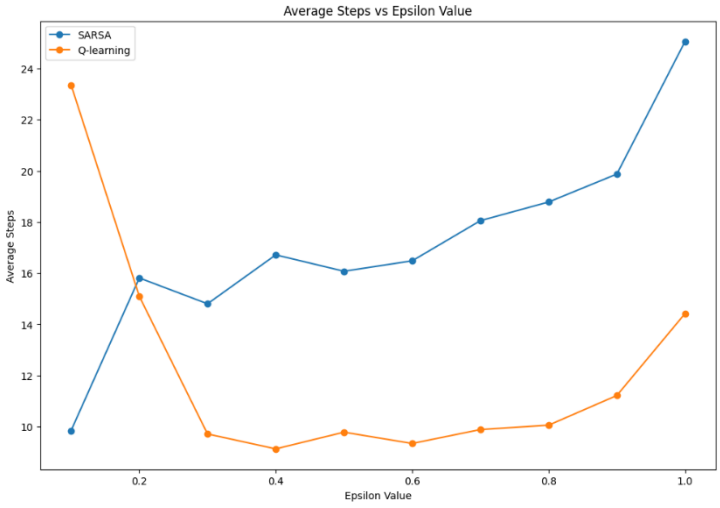
Comparison with Epsilon = 0.2, Epsilon = 0.6, and Epsilon = 1 for alpha = 0.3:

When compared to Epsilon values of 0.2 and 0.6, both SARSA and Q-learning with Epsilon set to 1 exhibited somewhat slower learning times (0.07 seconds for SARSA and 0.04 seconds for Q-learning) but notably lower average rewards (2.82 for SARSA and 3.52 for Q-learning). This suggests that despite the higher exploration rate with Epsilon set to 1, which led to slower learning, the resulting strategies were less effective. This was evidenced by more penalties incurred or less efficient navigation toward the goal compared to when using lower exploration rates.

Optimal Epsilon Analysis:

The optimal epsilon values for SARSA and Q-learning were determined based on their performance in terms of average rewards and the number of steps taken. SARSA achieved its highest average reward (9.605) with epsilon set to 0.1, highlighting a balanced approach that favors exploitation over exploration. This conservative strategy led to quicker convergence with fewer steps. In contrast, Q-learning reached its peak average reward (7.09) with epsilon set to 0.4, indicating that a slightly higher exploration rate was advantageous for discovering policies that yielded higher rewards, despite potentially requiring more initial exploration steps. This comparison illustrates the trade-off in reinforcement learning between strategies that exploit known information (lower epsilon values like 0.1 for SARSA) for faster convergence versus those that explore new possibilities (higher values like 0.4 for Q-learning) to potentially find more rewarding policies, even if more exploration is initially needed.





Effect of Changing Alpha on Exploration and Learning Efficiency:

Adjusting the exploration parameter, epsilon, has varying effects on SARSA and Q-learning depending on the value of alpha. With alpha set to 0.1, higher epsilon values resulted in quicker learning times (SARSA: 0.08 seconds, Q-learning: 0.05 seconds) but lower average rewards (SARSA: 4.48, Q-learning: 4.685). In contrast, with alpha set to 0.3, learning was slower (SARSA: 0.07 seconds, Q-learning: 0.04 seconds) and rewards were significantly lower (SARSA: 2.82, Q-learning: 3.52) when epsilon was set to 1. This illustrates that while increasing exploration accelerates learning, it also compromises the optimality of policies, potentially leading to more penalties or less effective navigation strategies.

Effect of Epsilon Values on Training:

- **SARSA:** When epsilon is low ($\varepsilon = 0.2$), algorithm explores less and focuses more on exploiting known actions, which helps it converge faster but might mean it misses out on finding better actions. With a medium epsilon ($\varepsilon = 0.6$), It strikes a balance between exploring new possibilities and exploiting known strategies. A high epsilon ($\varepsilon = 1$) leads algorithm to explore extensively, speeding up learning but potentially resulting in less optimal strategies initially due to taking more risks during exploration.
-
- **Q-learning:** Similarly, when epsilon is low ($\varepsilon = 0.2$), it emphasizes exploiting known strategies to quickly refine them. A medium epsilon ($\varepsilon = 0.6$) in balances between exploring new options and exploiting known strategies, which can lead to discovering better overall strategies. A high epsilon ($\varepsilon = 1$) encourages algorithm to explore extensively, facilitating faster learning but potentially resulting in less effective strategies in the early stages.

Optimal Value for Epsilon:

Based on the experiments, SARSA performed optimally around $\boldsymbol{\varepsilon = 0.2}$, striking a good balance between exploring new options and exploiting known ones. For Q-learning, around $\boldsymbol{\varepsilon = 0.4}$ was optimal, allowing for sufficient exploration to find better strategies while still exploiting known good actions.

V. Conclusion

In summary, this report thoroughly examined SARSA and Q-learning algorithms in a deterministic grid world setting, aiming to understand their performance under varying exploration parameters. SARSA demonstrated robust efficiency with $\boldsymbol{\varepsilon} = 0.2$, achieving quick convergence and a high average reward of **9.605**, albeit with slightly more steps (**14**) to the goal. In contrast, Q-learning thrived with epsilon = 0.4, striking a balance between exploration and exploitation to converge swiftly in 0.03 seconds with (**8**) steps and an average reward of **7.09**. Higher epsilon values ($\boldsymbol{\varepsilon} = 1$) expedited learning but resulted in lower average rewards due to increased exploration risks. These findings highlight the pivotal role of epsilon in influencing learning dynamics: SARSA with an epsilon value of approximately **0.2** leans towards exploitation, enhancing efficiency in policy development. On the other hand, Q-learning benefits from an epsilon value around **0.4**, promoting a balanced exploration-exploitation trade-off crucial for uncovering optimal strategies. This study provides significant insights into parameter tuning in reinforcement learning, offering practical guidance for optimizing algorithmic performance in real-world scenarios.