Reinforcement Learning in Complex Route Navigation and Spatial Decision Making

Indrajeet Roy
Department of Electrical and Computer Engineering, Northeastern University
roy.i@northeastern.edu

Source code github repo Click Here

State Space

$$\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_{248}\}$$

$$\mathcal{A} = \{U, D, R, L\}$$

$$\begin{aligned}
&\text{For } a = R \text{ (Right)}: && P(s_{\text{right}} \mid s, R) = 1 - p, && P(s_{\text{left}} \mid s, R) = \frac{p}{3}, && P(s_{\text{up}} \mid s, R) = \frac{p}{3}, && P(s_{\text{down}} \mid s, R) = \frac{p}{3} \\
&\text{For } a = U \text{ (Up)}: && P(s_{\text{up}} \mid s, U) = 1 - p, && P(s_{\text{down}} \mid s, U) = \frac{p}{3}, && P(s_{\text{left}} \mid s, U) = \frac{p}{3}, && P(s_{\text{right}} \mid s, U) = \frac{p}{3} \\
&\text{For } a = D \text{ (Down)}: && P(s_{\text{down}} \mid s, D) = 1 - p, && P(s_{\text{up}} \mid s, D) = \frac{p}{3}, && P(s_{\text{left}} \mid s, D) = \frac{p}{3}, && P(s_{\text{right}} \mid s, D) = \frac{p}{3} \\
&\text{For } a = L \text{ (Left)}: && P(s_{\text{left}} \mid s, L) = 1 - p, && P(s_{\text{right}} \mid s, L) = \frac{p}{3}, && P(s_{\text{up}} \mid s, L) = \frac{p}{3}, && P(s_{\text{down}} \mid s, L) = \frac{p}{3}
\end{aligned}$$

$$r(s, a, s') = \begin{cases} 200 - 1 & \text{if } s' \text{ is the goal state and taking action } a \\ -5 - 1 & \text{if } s' \text{ is an oil state and taking action } a \\ -10 - 1 & \text{if } s' \text{ is a bump state and taking action } a \\ -1 & \text{for taking action } a \end{cases}$$

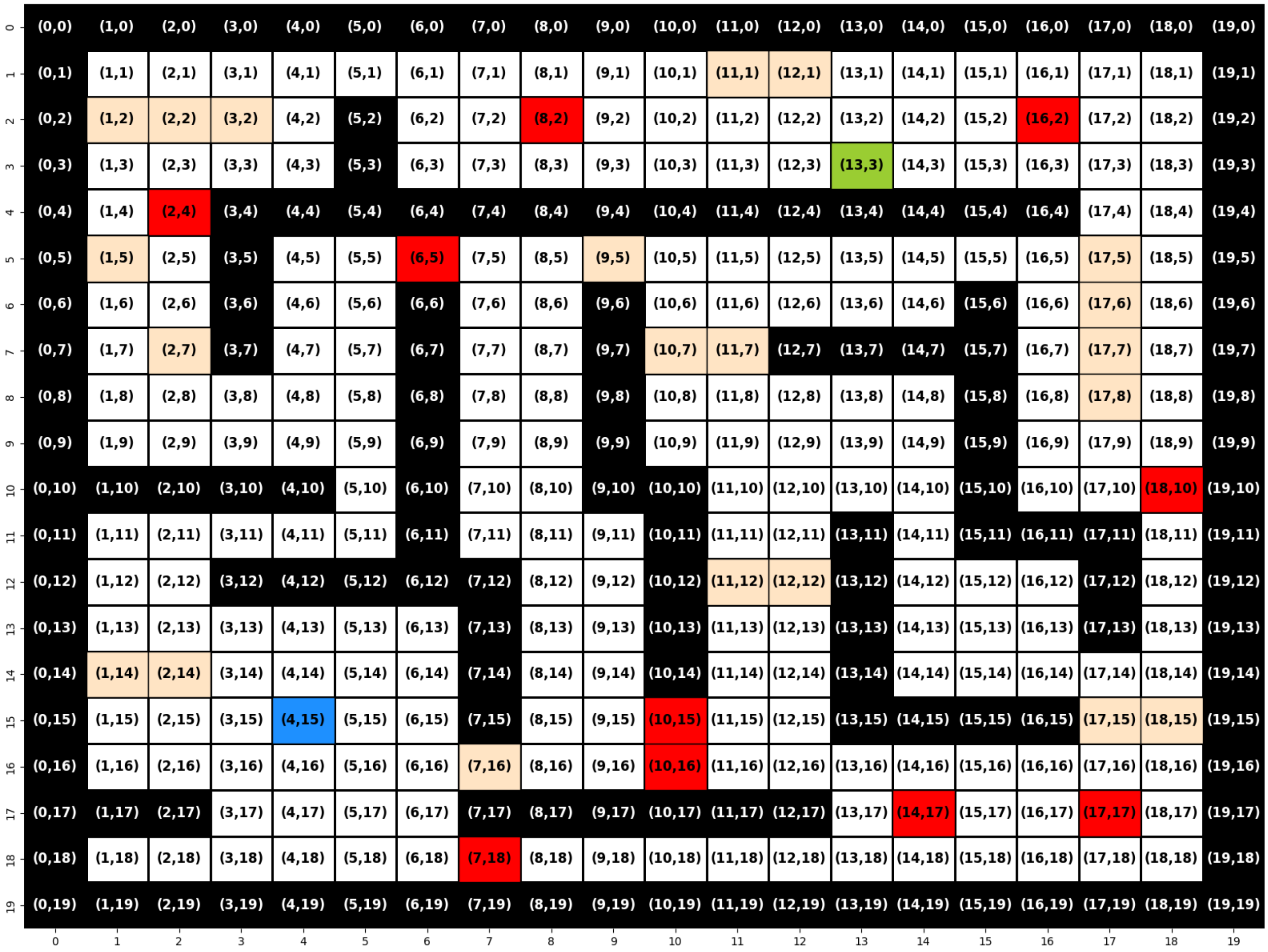

Figure 1: Maze consisting of walls, bumps and oils, goal and starting point

Q-Learning

In 10 independent runs of Q-Learning for navigation, each with parameters $\epsilon = 0.1$, $\alpha = 0.3$, and $\gamma = 0.95$, over a total of 1,000 episodes and a maximum episode length of 1,000, a successful path from start to goal was obtained in 10 runs upon the termination of learning.

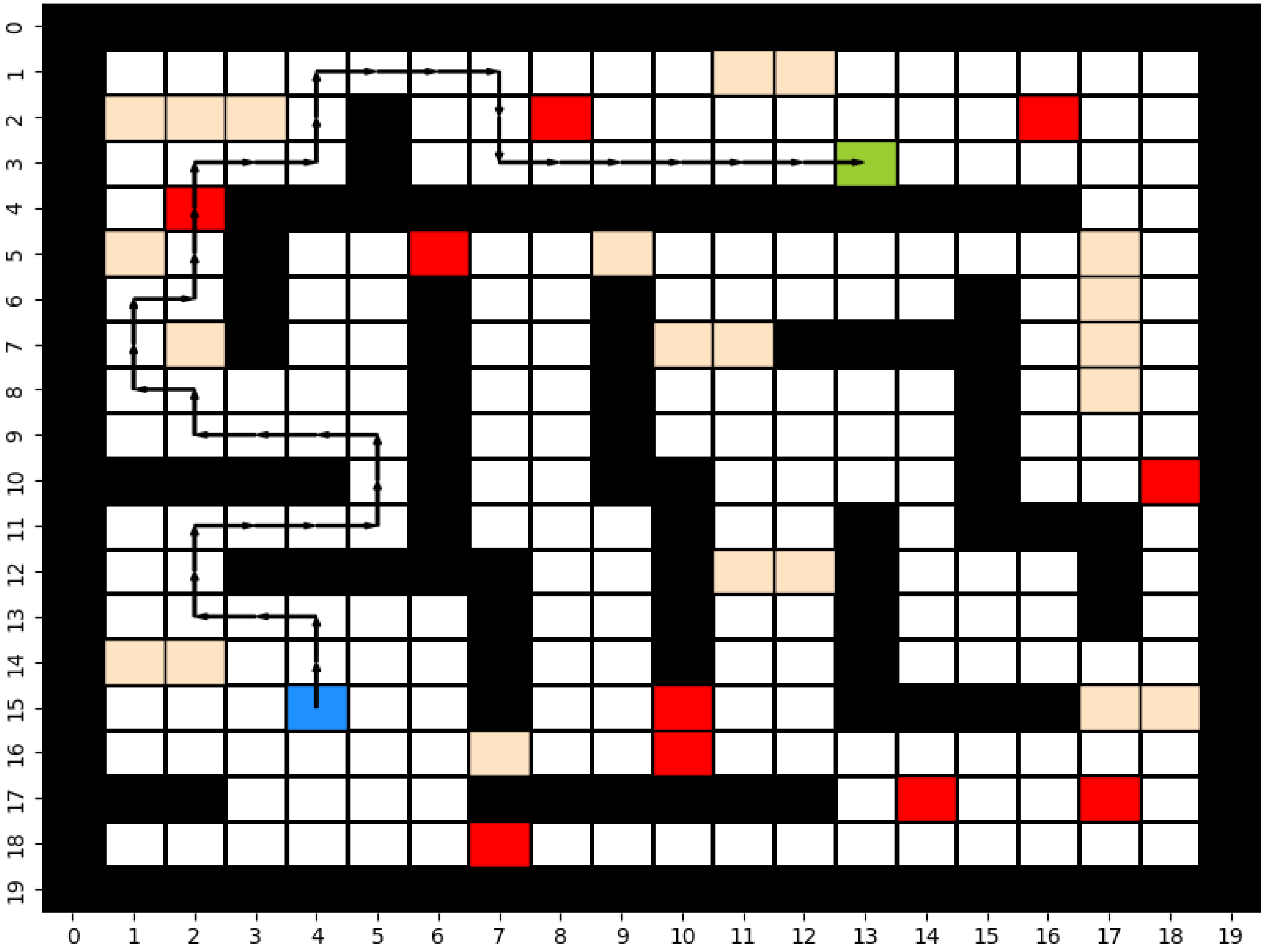

Figure 2: Optimal path from start to goal for one of the 10 independent runs for Q-Learning

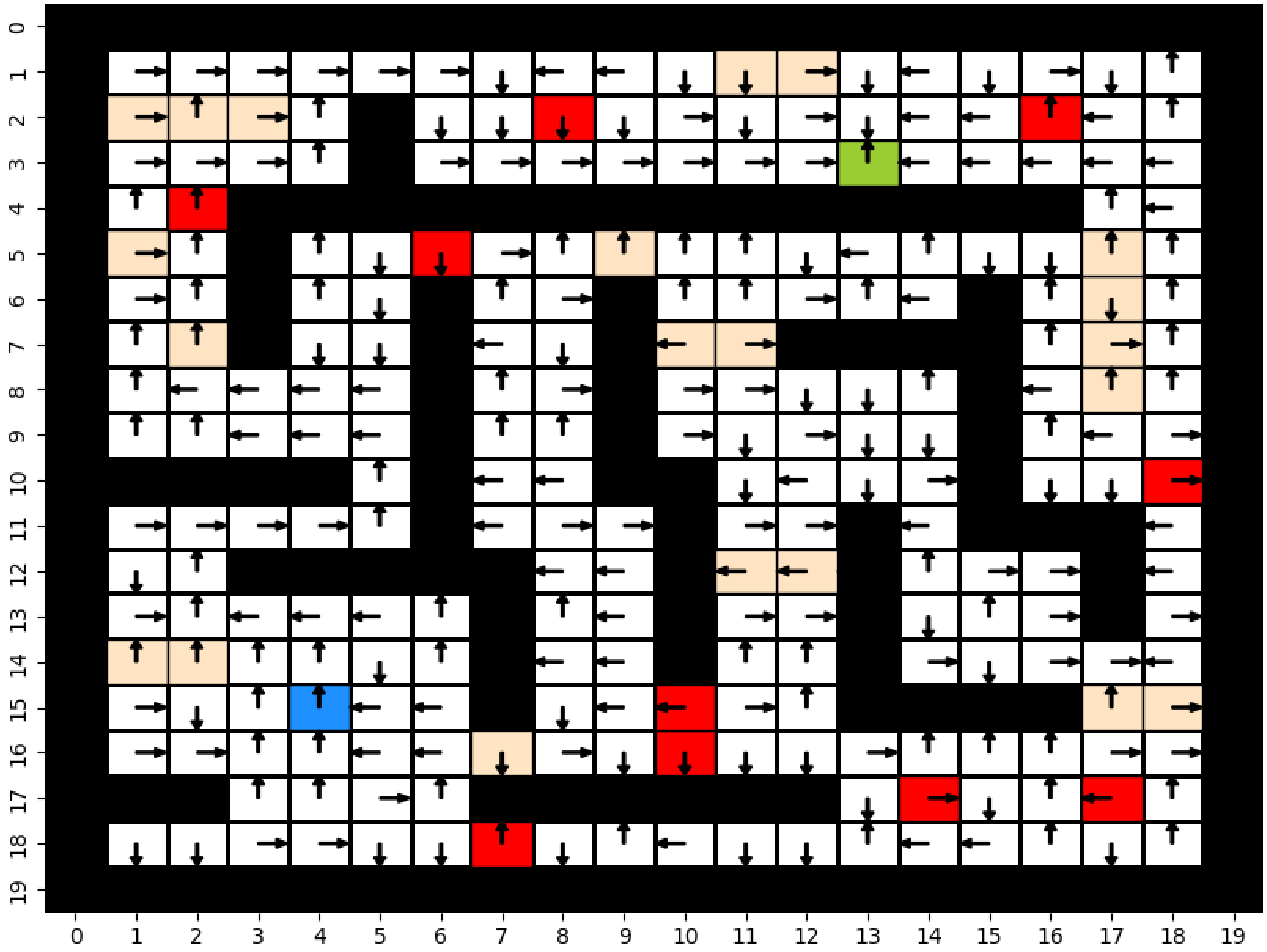

Figure 3: Optimal policy from start to goal for one of the 10 independent runs for Q-Learning

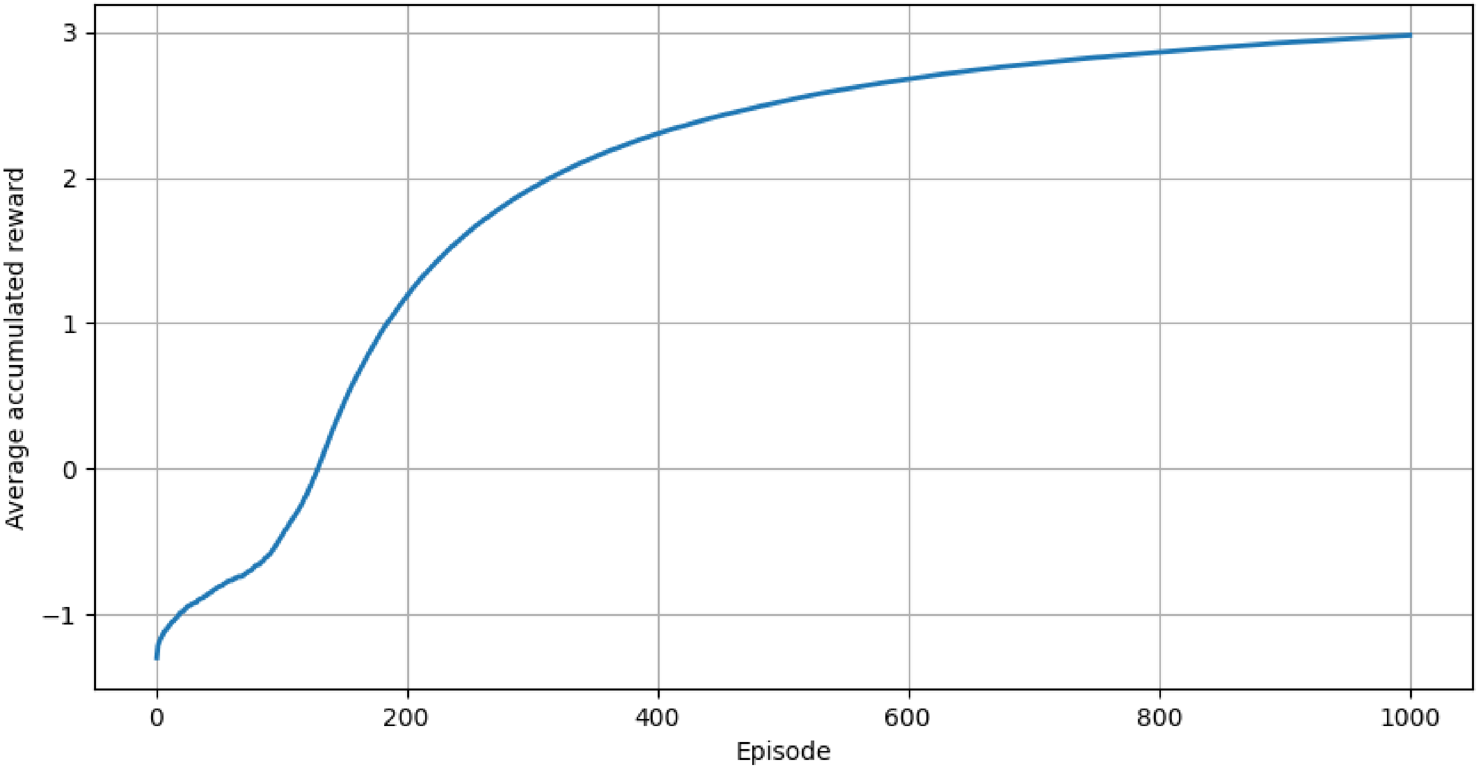

Figure 4: Average accumulated reward (in 10 independent runs) w.r.t episode number for Q-Learning

SARSA

In 10 independent runs of SARSA for navigation, each with parameters $\epsilon = 0.1$, $\alpha = 0.3$, and $\gamma = 0.95$, over a total of 1,000 episodes and a maximum episode length of 1,000, a successful path from start to goal was obtained in 10 runs upon the termination of learning.

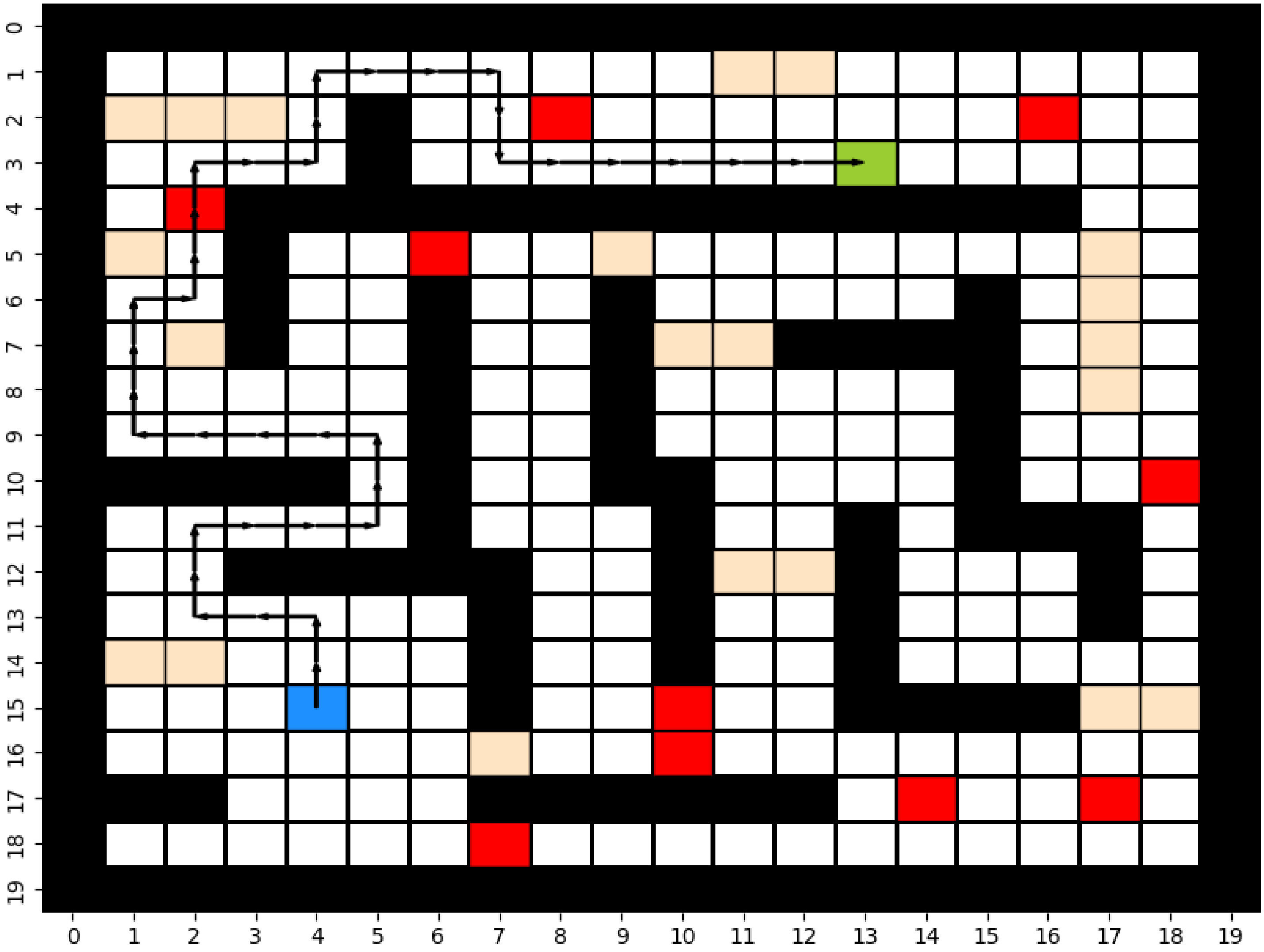

Figure 5: Optimal path from start to goal for one of the 10 independent runs for SARSA

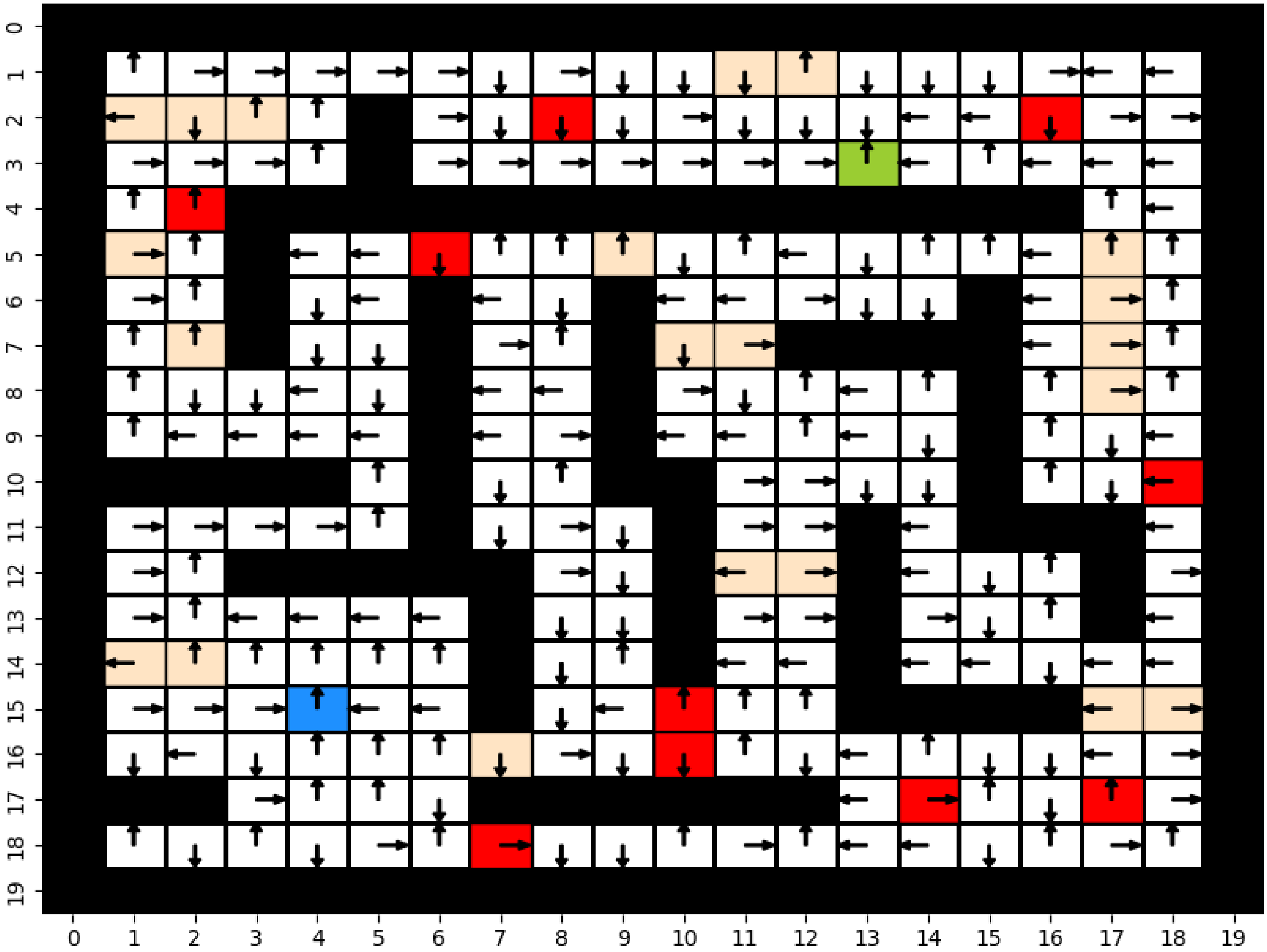

Figure 6: Optimal policy from start to goal for one of the 10 independent runs for SARSA

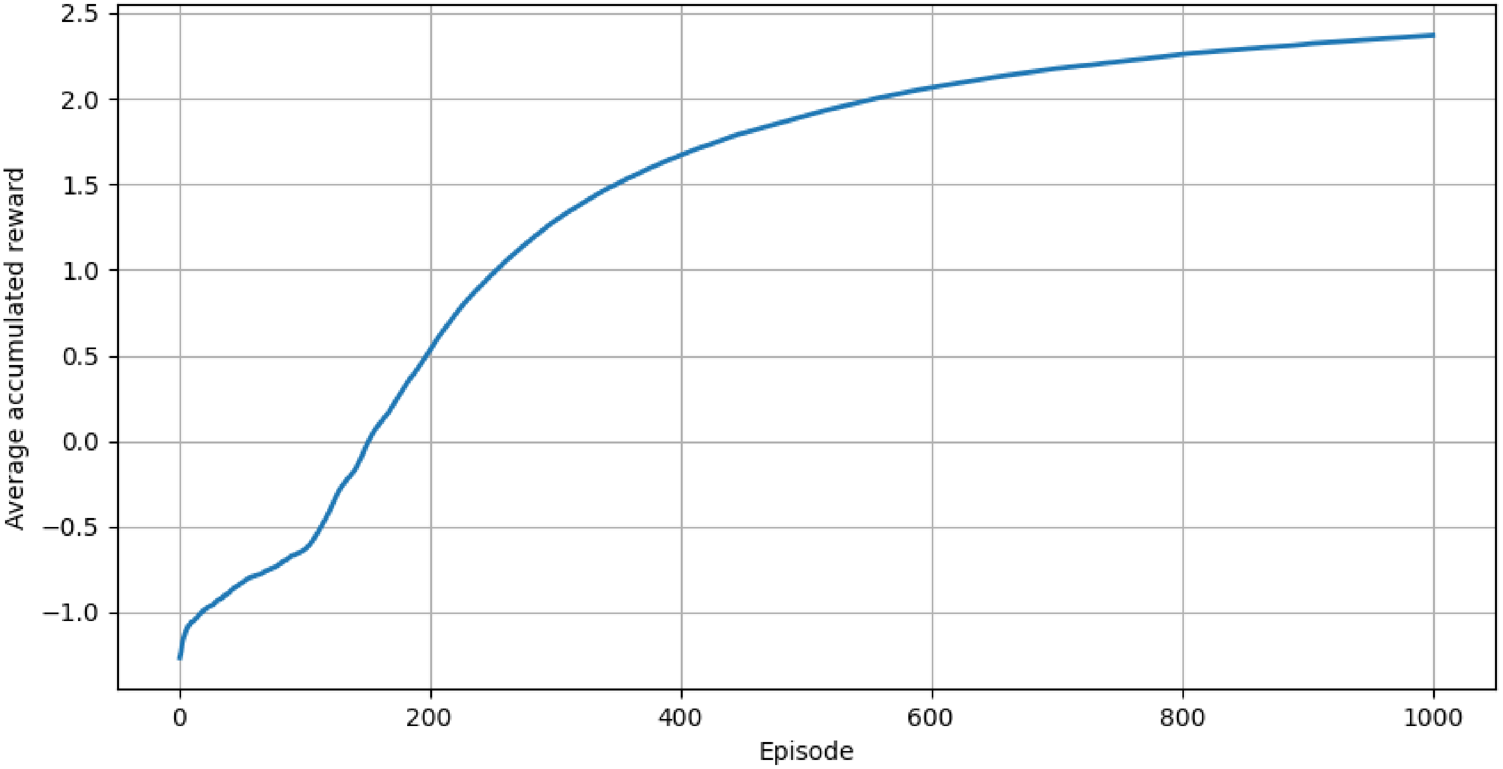

Figure 7: Average accumulated reward (in 10 independent runs) w.r.t episode number for SARSA

Actor-Critic

In 10 independent runs of Actor-Critic for navigation, each with parameters $\beta = 0.05$, $\lambda = 0.9$, $\alpha = 0.3$, and $\gamma = 0.95$, over a total of 1,000 episodes and a maximum episode length of 1,000, a successful path from start to goal was obtained in 10 runs upon the termination of learning.

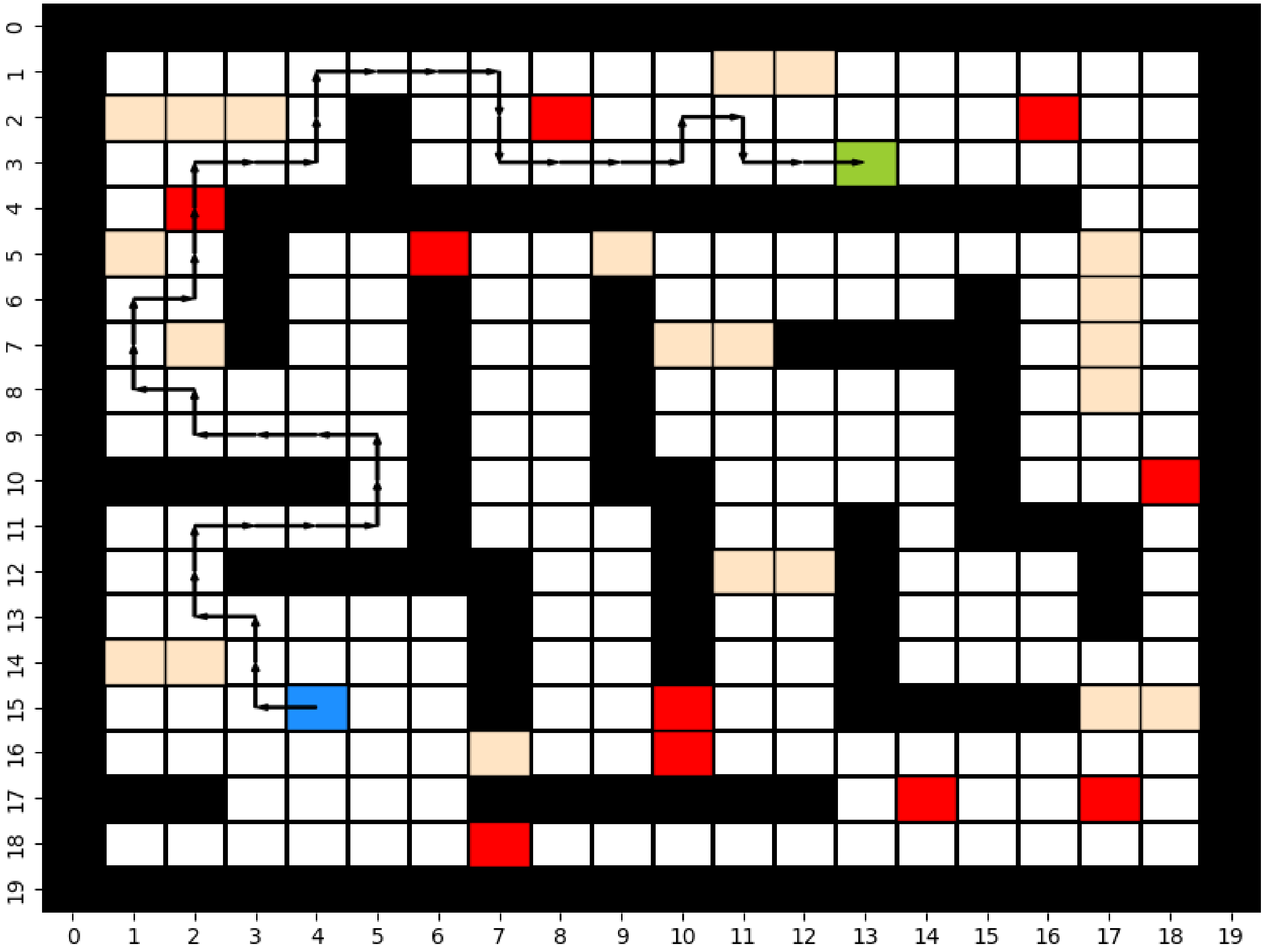

Figure 8: Optimal path from start to goal for one of the 10 independent runs for Actor-Critic

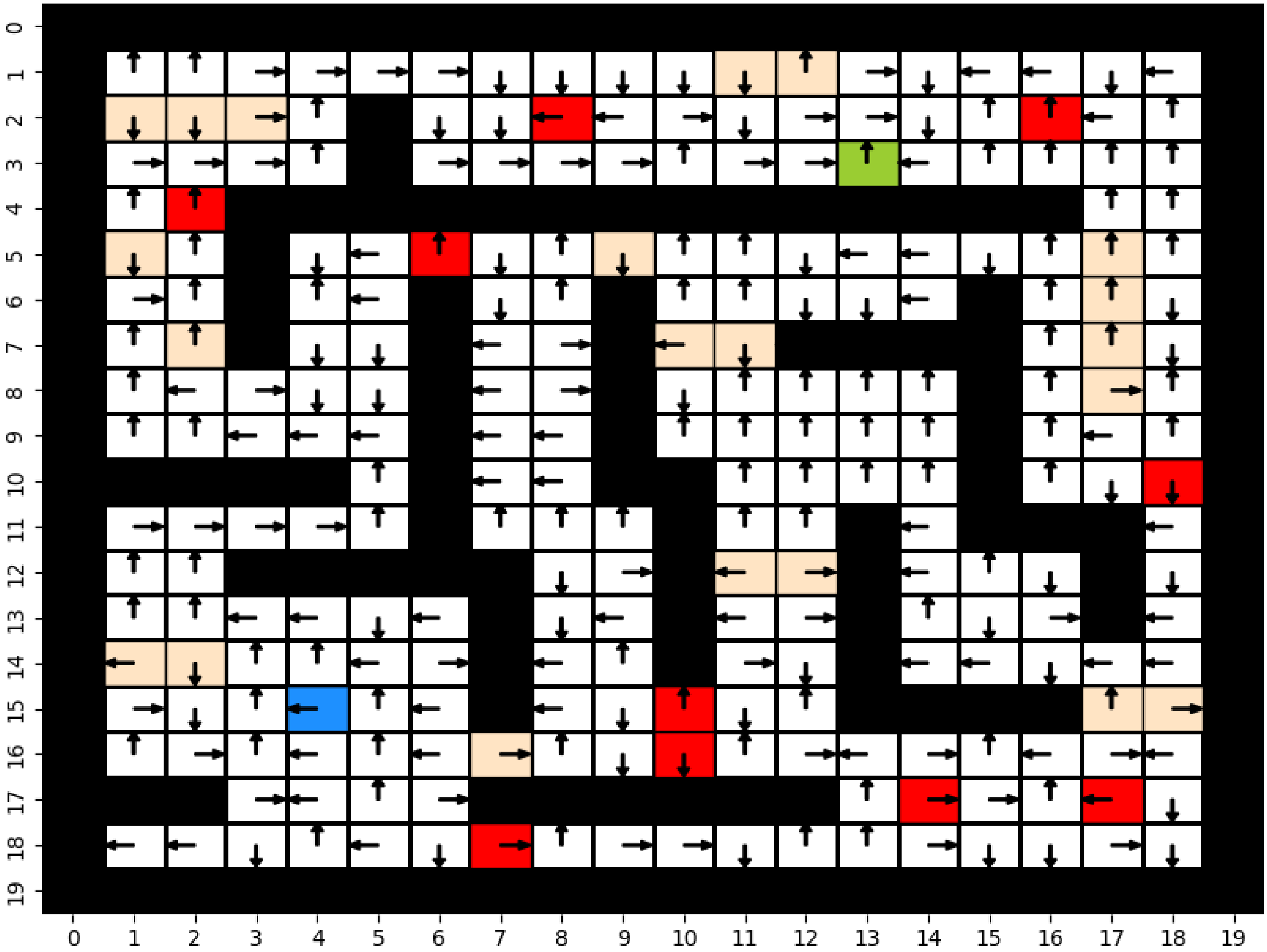

Figure 9: Optimal policy from start to goal for one of the 10 independent runs for Actor-Critic

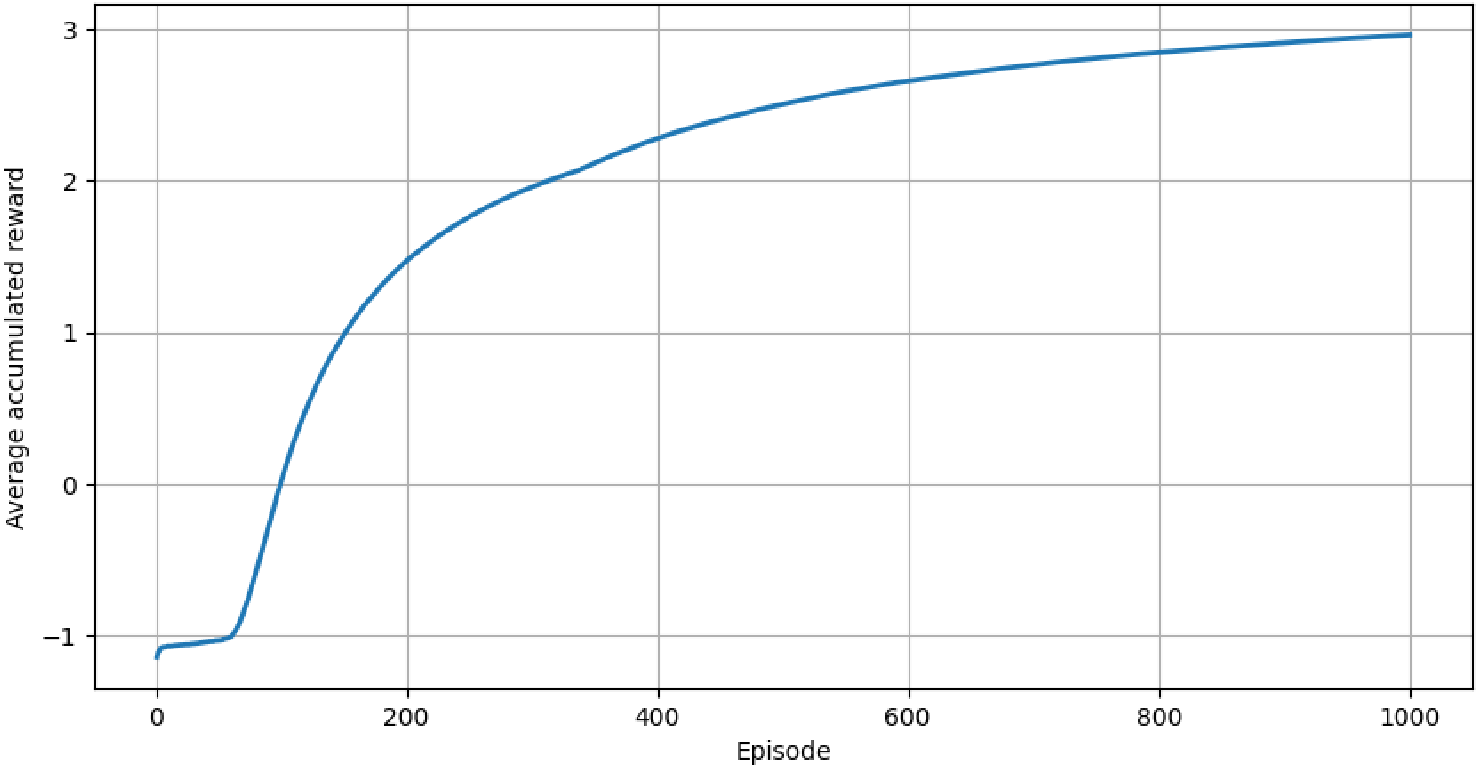

Figure 10: Average accumulated reward (in 10 independent runs) w.r.t episode number for Actor-Critic

Analysis

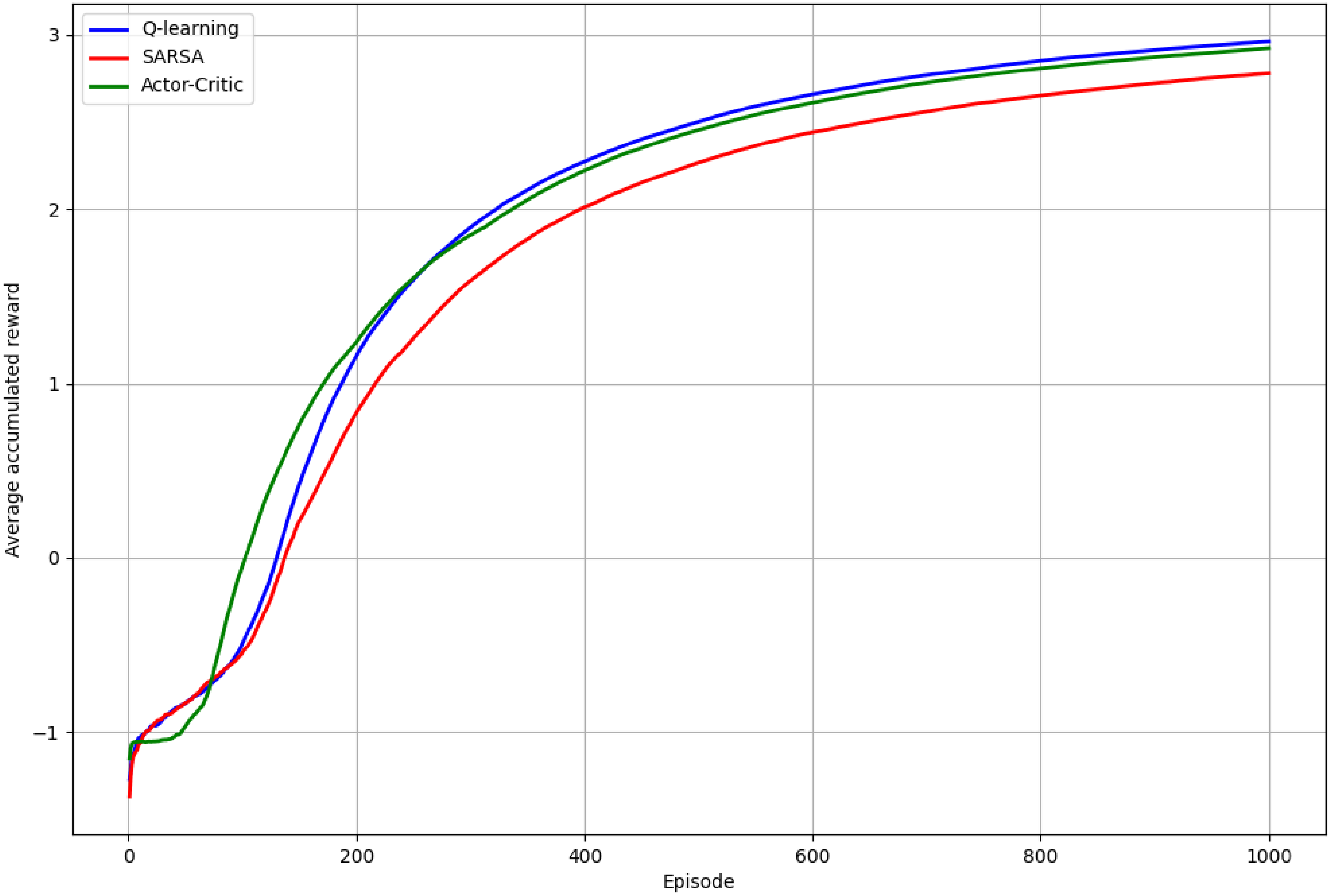

Figure 11: Average accumulated reward with respect to the episode number for all algorithms

Q-learning is an off-policy algorithm that estimates the value of the optimal policy by learning the action-value function, independent of the agent's current actions, allowing the algorithm to determine the best policy by learning from both actual and hypothetical actions, making it effective in the stochastic maze environment.From the graph it can be observed,during the initial learning phase (episodes 0-200), the Q-learning curve exhibits a rapid increase in accumulated reward.This steep ascent in the early episodes highlights the Q-learning algorithm's effective exploration and ability to quickly learn from the environment.Q-learning updates its Q-values using the maximum expected future rewards, which facilitates efficient policy improvement early in the learning process.During the mid learning phase (episodes 200-600),the Q-learning curve begins to plateau, indicating that the algorithm is beginning to exploit the knowledge it has gained.The gradual leveling off suggests that the Q-learning algorithm is refining its policy towards the optimal policy, and the rate of learning new information is slowing down as it exploits acquired knowledge in comparison to exploration of the environment.During the late learning phase (episodes 600-1000),the Q-learning curve plateaus suggesting that the Q-learning algorithm has converged to an optimal policy with the maximized average accumulated reward. The final Q-learning curve highlights the algorithm's success in learning an optimal policy that effectively navigates the stochastic maze environment and maximizes accumulated rewards, balancing exploration and exploitation in an optimal way.

SARSA is an on-policy algorithm, where the algorithm learns from the actions it actually takes in comparison to hypothetical actions.SARSA learns the value of the policy it follows based on the actions taken.During the initial learning phase (episodes 0-200), the SARSA curve is less steep in comparison to the Q-learning curve, which highlights the algorithms on-policy approach as the algorithm updates Q-values based on the action that is actually taken in comparison to the the maximum reward action, incorporating exploration into its learning. This exploration approach contributes to a more conservative trajectory, as the algorithm directly experiences and learns from the consequences such as penalties resulting in relatively less accumulated reward.The less steep curve also highlights the SARSA algorithms preference to prioritize safety in comparison to reward, which overlook potentially beneficial exploratory actions that carry higher risk but also the possibility of greater rewards.During the mid learning phase (Episode 200-600) the SARSA curve highlights the algorithms preference for safety resulting from updating policy based on the currently taken actions as the curve is slowly progressing toward higher rewards but the accumulated rewards are relatively less in comparison to Q-learning and Actor-Critic.During the late learning phase (Episode 600-1000),the SARSA curve begins to plateau and continues to remain the lowest accumulated reward among the three algorithms suggesting the algorithms steady and risk-averse progression, potentially indicating a more conservative and potentially safer policy, but one that is less optimized for maximizing rewards in the maze environment with stochasticity.

Actor-Critic algorithm utilizes the benefits of both on-policy and off-policy approaches by combining value function approximation (critic) with policy optimization (actor) to achieve a balance between the flexibility of policy-based method and the stability and efficiency of value-based method.During the initial learning phase (Episode 0-200), the Actor-Critic curve is similar to the Q-learning curve highlighting rapid increase in the accumulated reward. This indicates the actor's capability to quickly adapt policy based on the critic's accurate value assessment of policy's performance, enabling the actor to adjust the policy efficiently, effectively balancing exploitation and exploration of the environment.During the mid learning phase (Episode 200-600), the curve highlights a decrease in performance indicating sub optimal adjustments to the policy or limited improvement of the value function.During the late learning phase (Episode 600-1000), the Actor-Critic curve plateaus below the Q-learning curve, indicating that the algorithm method has found a relatively stable policy, but in it is not the optimal policy for maximizing rewards in the stochastic maze environment.The sub-optimal policy convergence could be based on the stochasticity in the environment effecting the feedback loop between Actor and Critic components.In a stochastic environment, the actor may not always get consistent feedback on the best actions to take, which may lead to less optimal policy convergence.