EECE5698 – Project 3

Assignment Source code github repo Click Here

Problem 1.

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

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

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

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

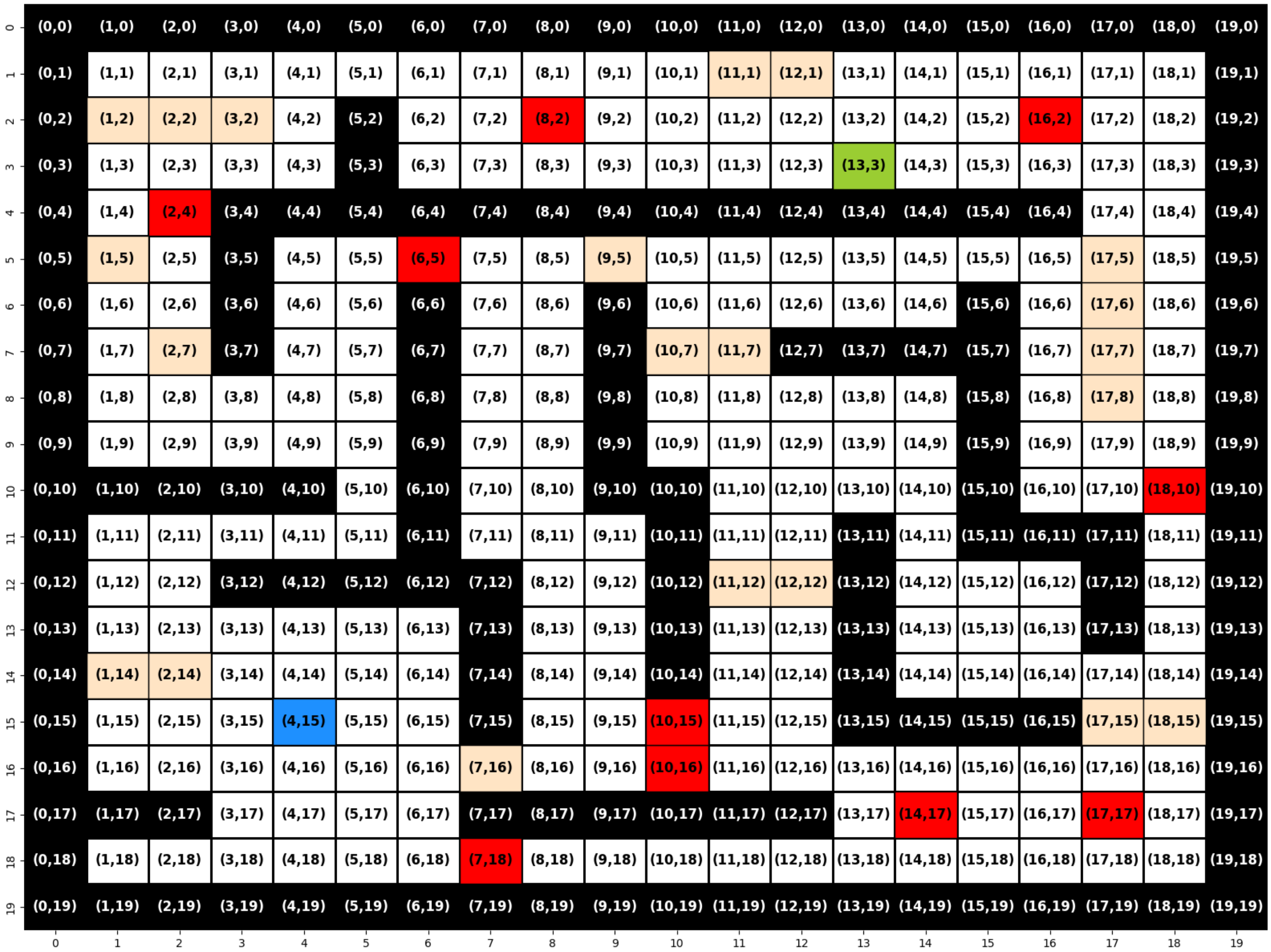

Figure 1: Maze consisting of walls, bumps and oils, goal and starting point

Q-Learning

a.

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.

b.

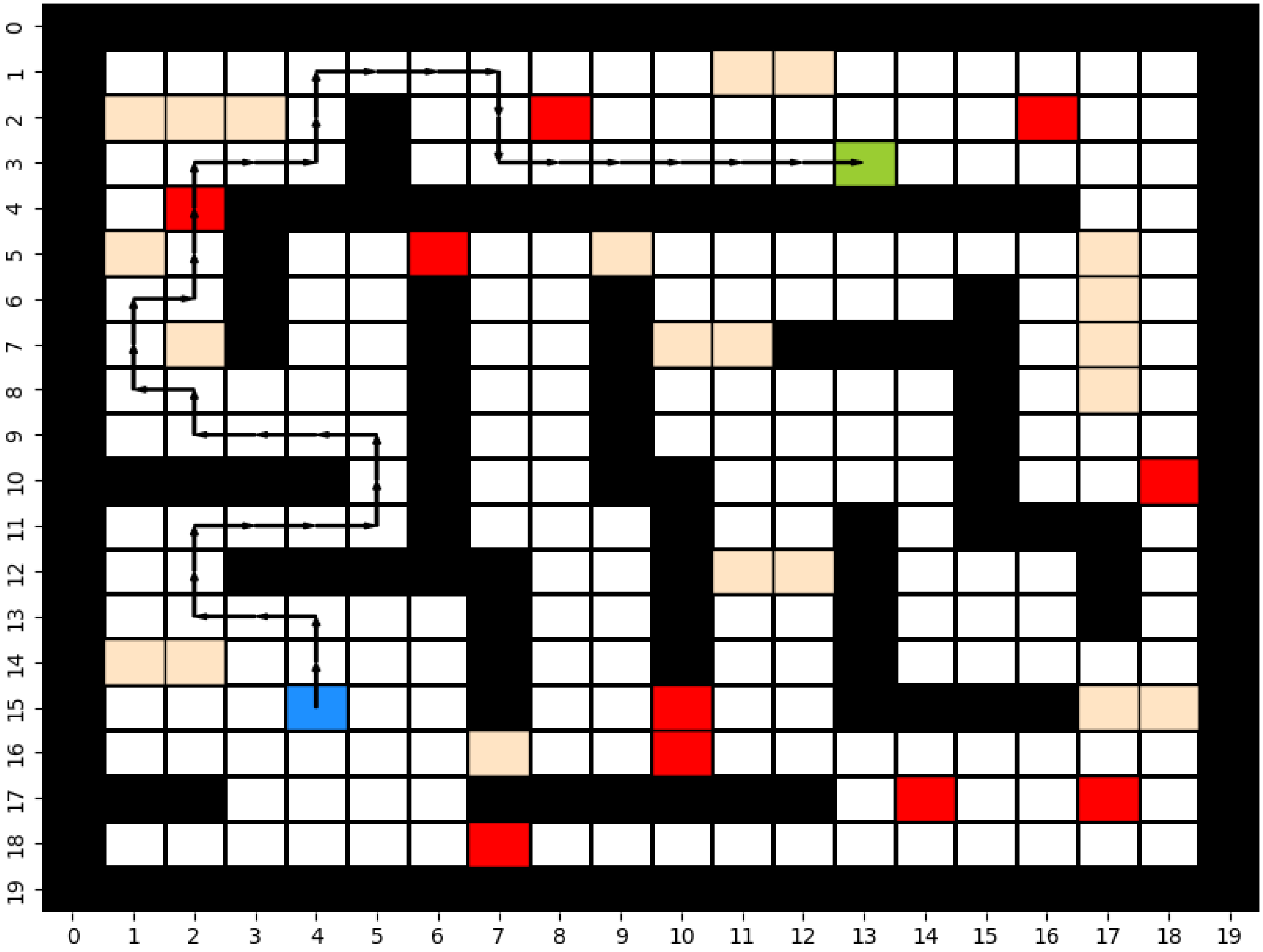

Figure 2: Optimal path from start to goal for one of the 10 independent runs for Q-Learning

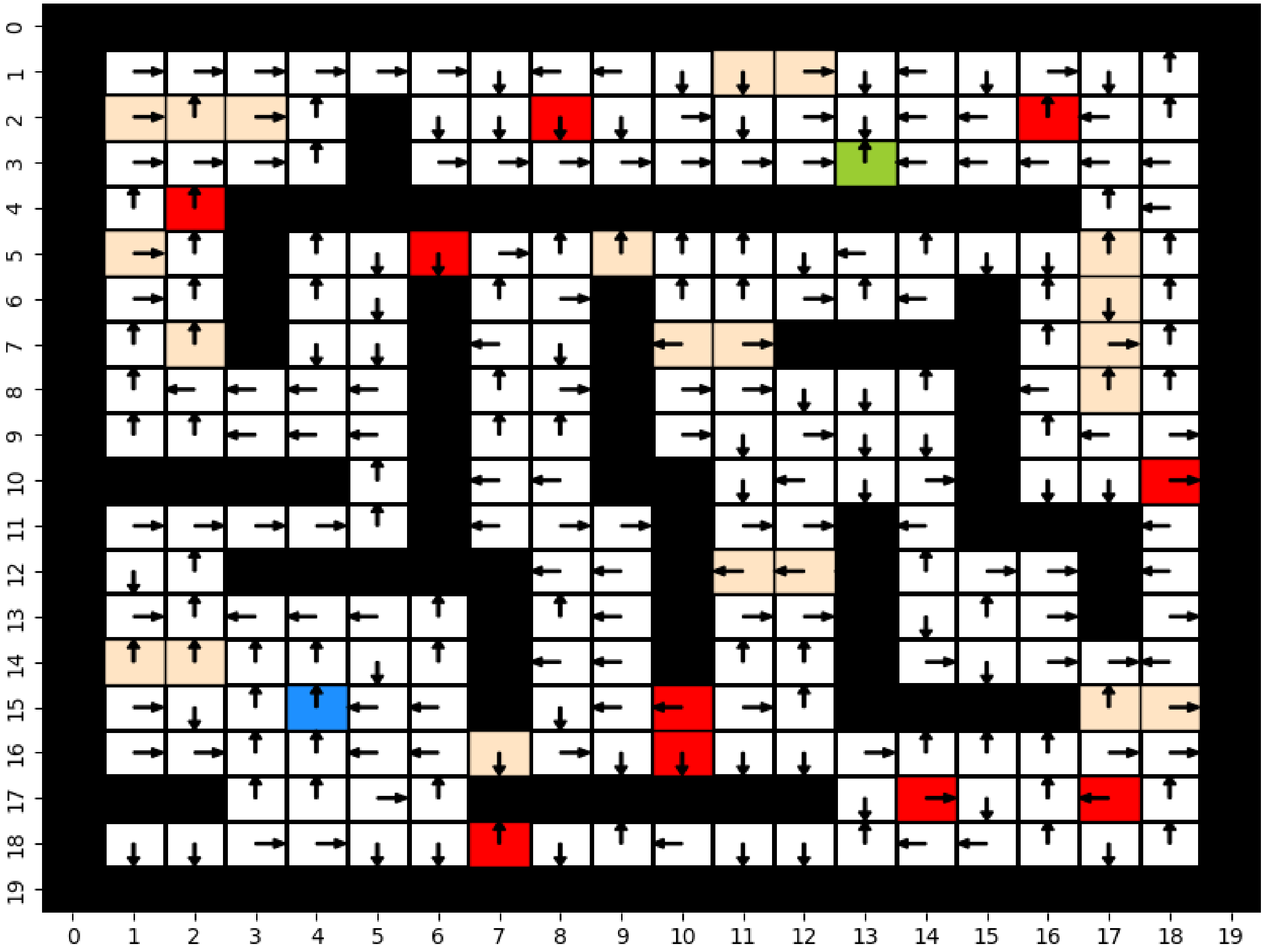

Figure 3: Optimal policy from start to goal for one of the 10 independent runs for Q-Learning

c.

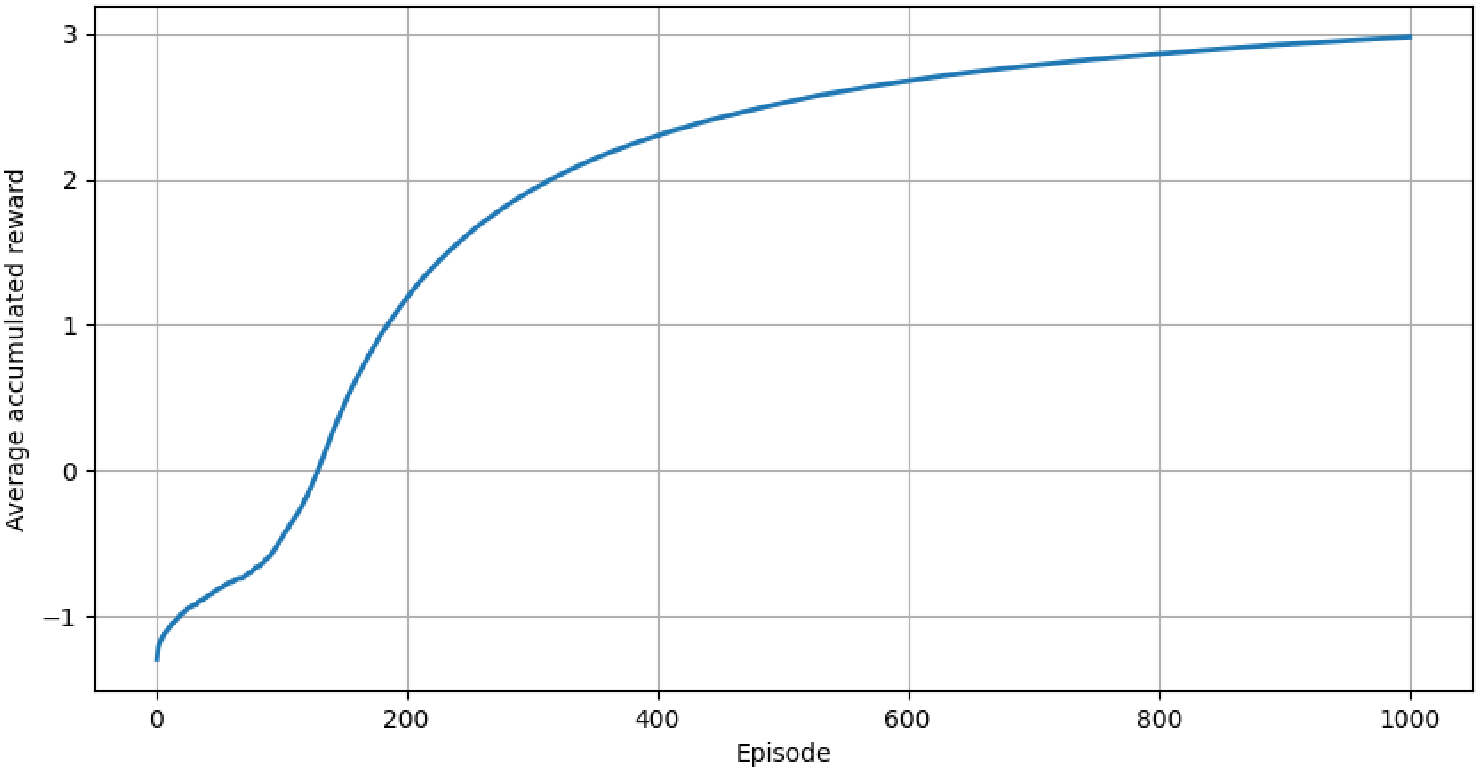

Figure 4: Average accumulated reward (in 10 independent runs) w.r.t episode number for Q-Learning

SARSA

a.

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.

b.

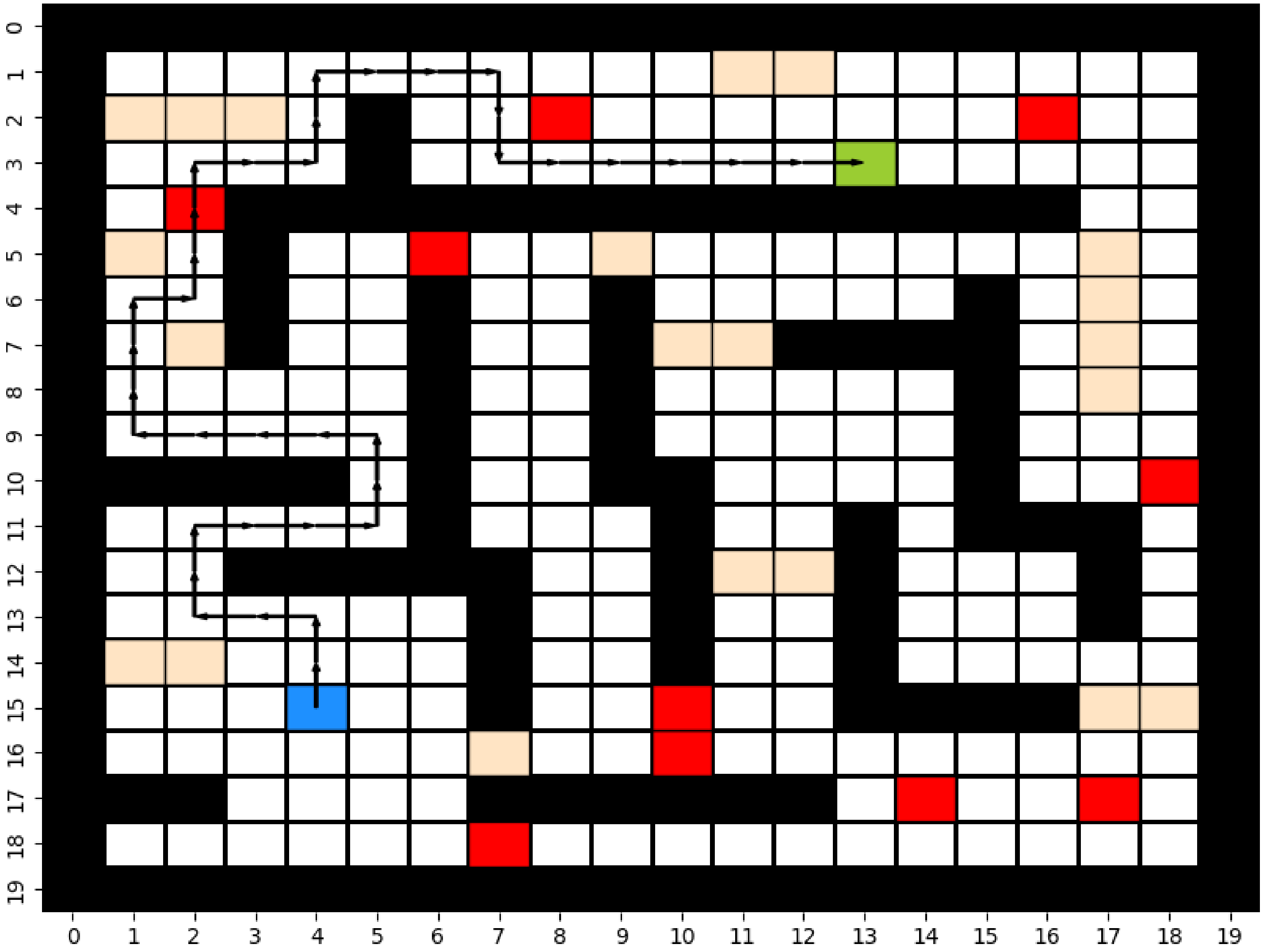

Figure 5: Optimal path from start to goal for one of the 10 independent runs for SARSA

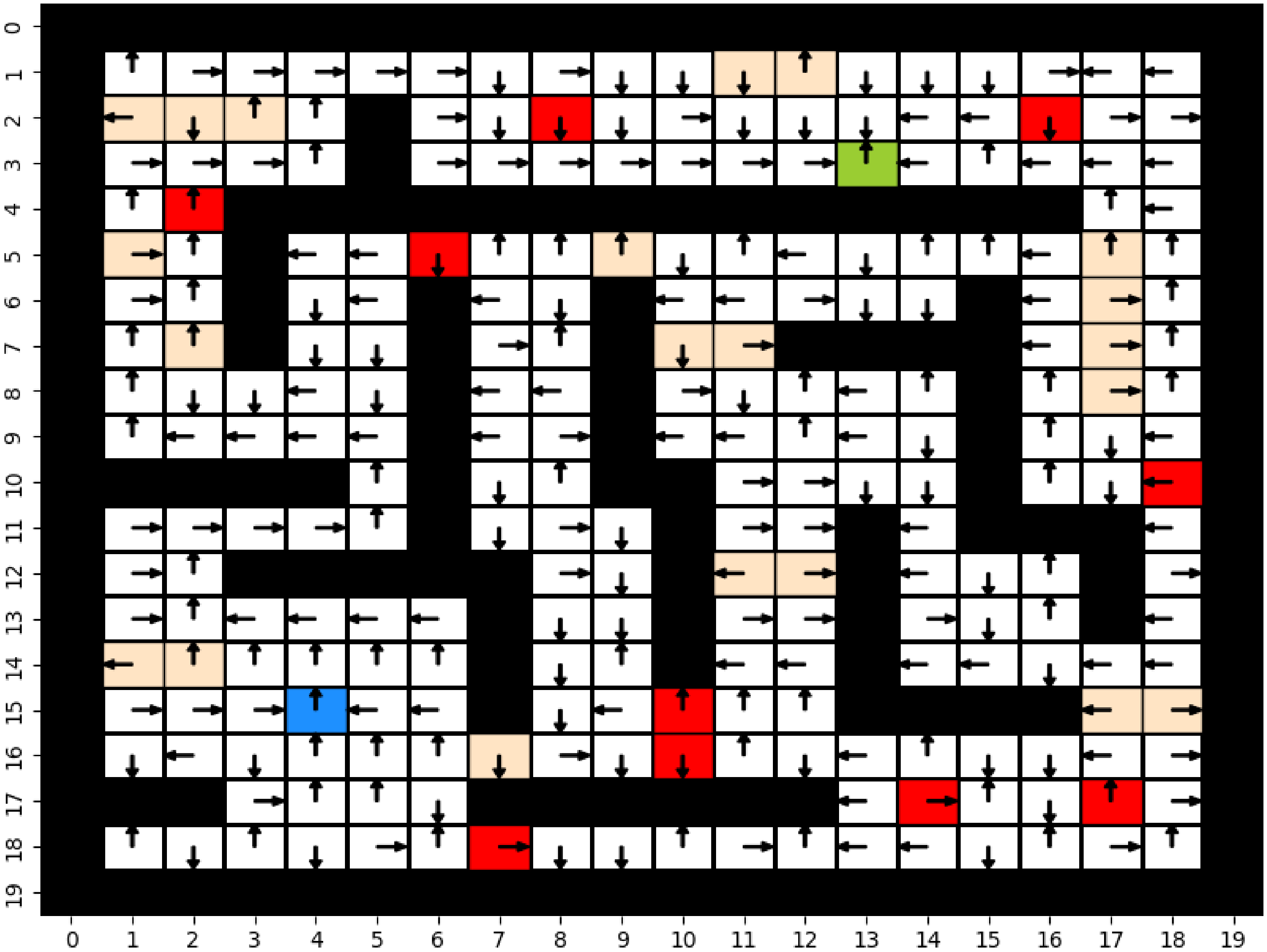

Figure 6: Optimal policy from start to goal for one of the 10 independent runs for SARSA

c.

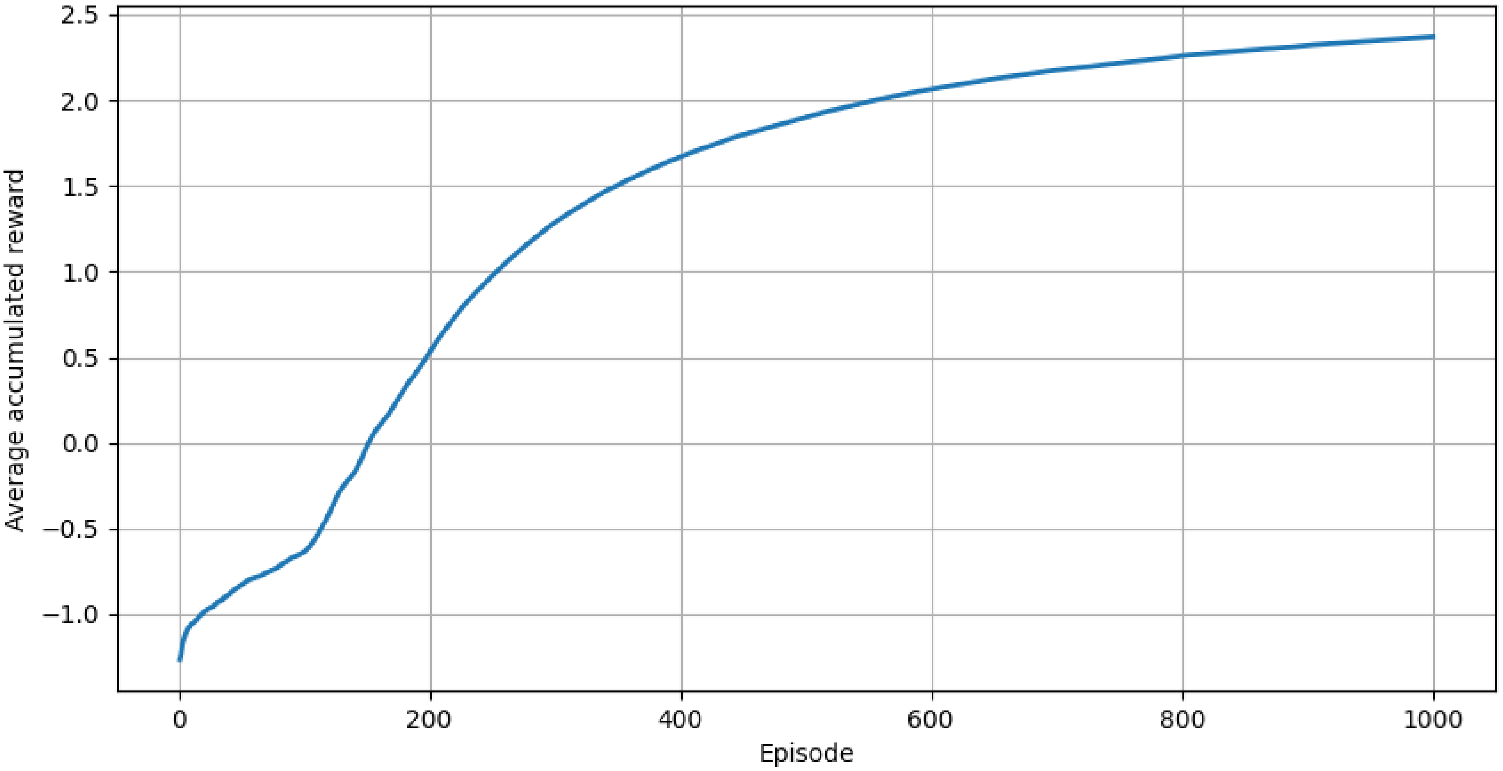

Figure 7: Average accumulated reward (in 10 independent runs) w.r.t episode number for SARSA

Actor-Critic

a.

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.

b.

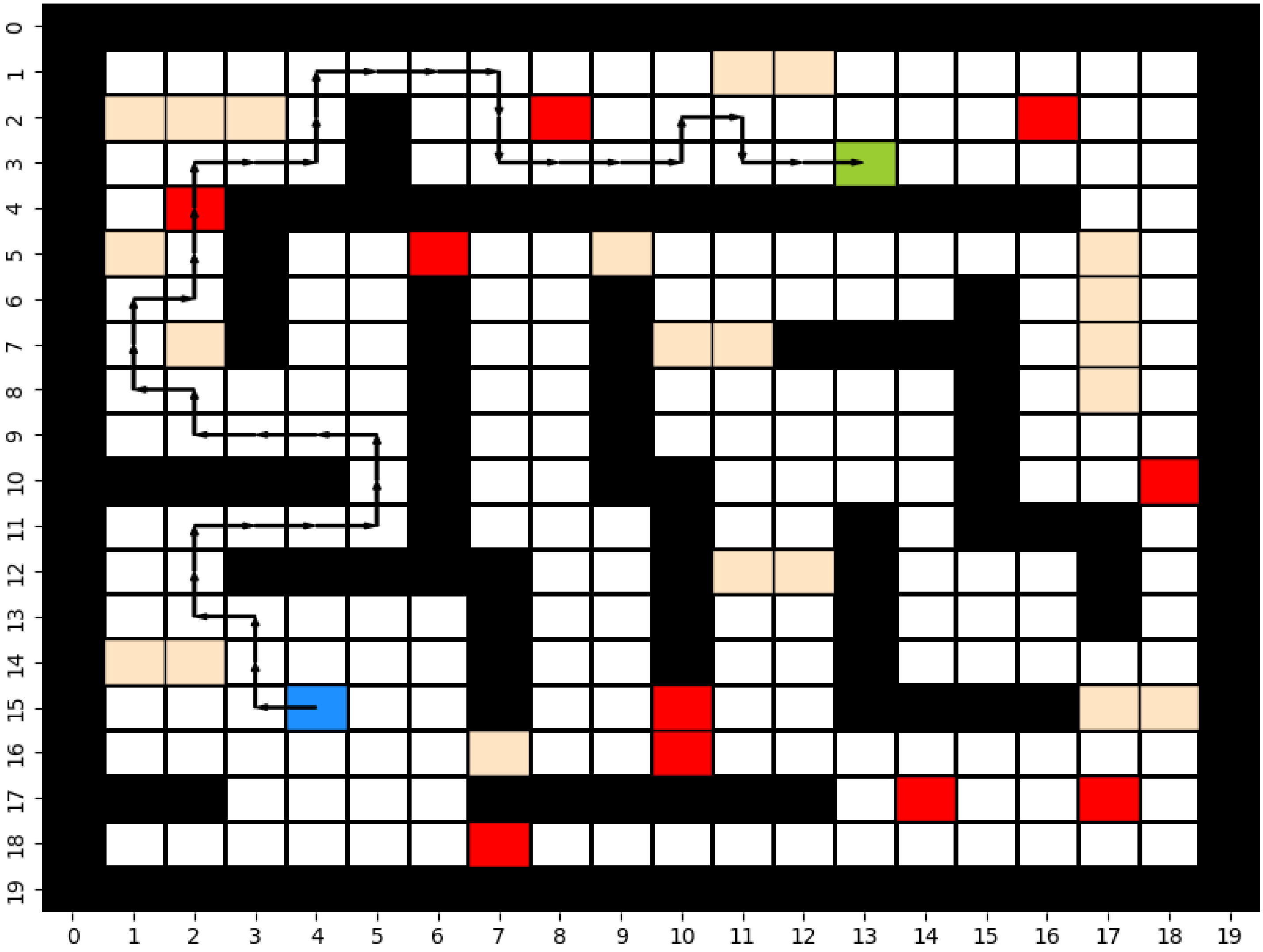

Figure 8: Optimal path from start to goal for one of the 10 independent runs for Actor-Critic

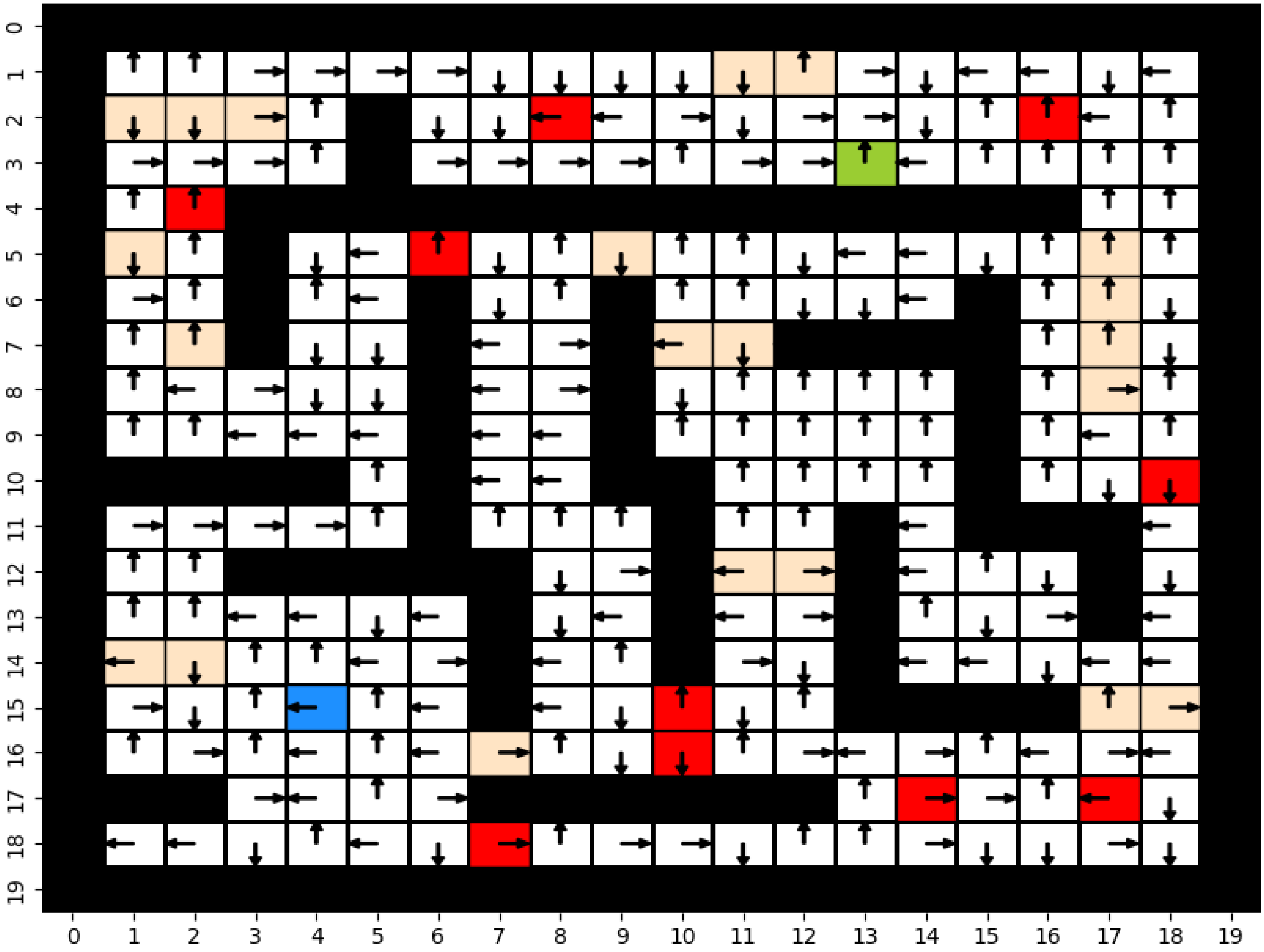

Figure 9: Optimal policy from start to goal for one of the 10 independent runs for Actor-Critic

c.

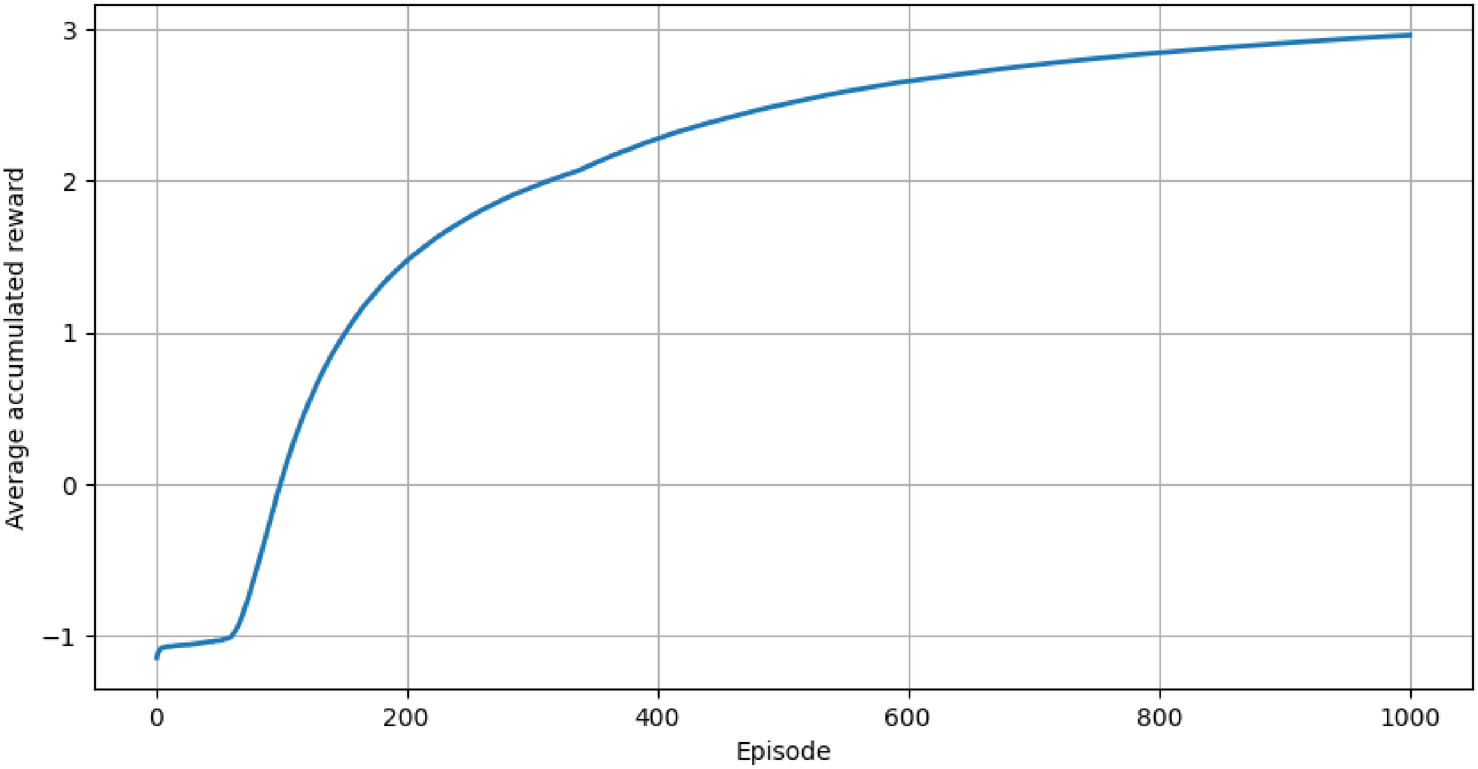

Figure 10: Average accumulated reward (in 10 independent runs) w.r.t episode number for Actor-Critic

d.

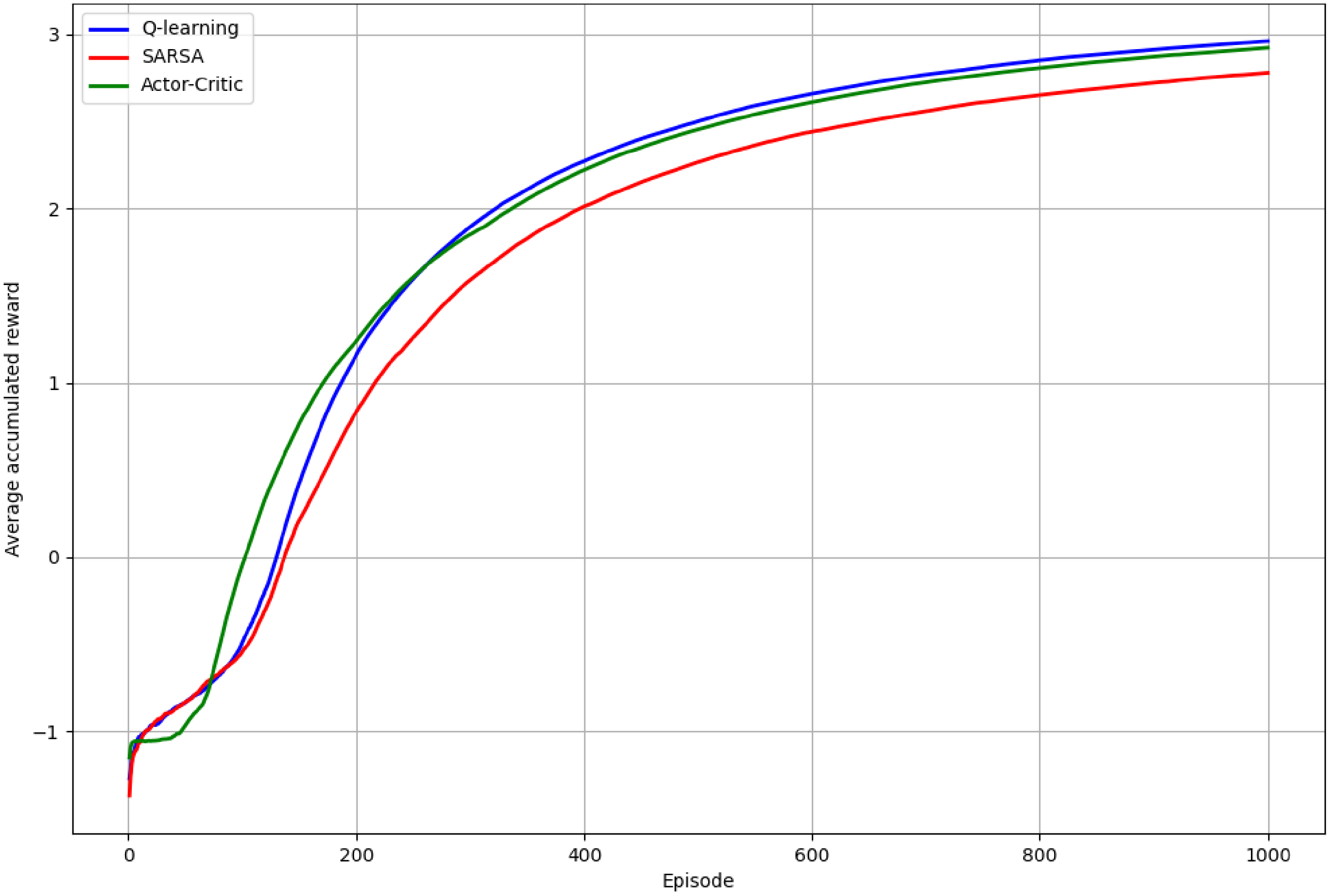

Figure 11: Average accumulated reward with respect to the episode number for all algorithms

Analysis

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.

Problem 2.

MDP for the p53-Mdm2 negative feedback loop network consists of 4 components/genes, represented in the state vector $\mathbf{s}_k = [\text{No Action}, \text{p53}, \text{Wpi}, \text{MDM2}]^T$. At any time, the state value of each gene can be: 0 (OFF) and 1 (ON).

$$\mathcal{S} = \left\{[0,0,0,0]^T, [0,0,0,1]^T, [0,0,1,0]^T, [0,0,1,1]^T, \ldots, [1,1,1,0]^T, [1,1,1,1]^T\right\}$$

$$\mathcal{A} = \left\{[0,0,0,0]^T, [0,1,0,0]^T, [0,0,1,0]^T, [0,0,0,1]^T\right\}$$

The state transition probability equation is defined as:

$$\mathbf{s}_k = \mathbf{C} \cdot (\mathbf{s}_{k-1} \oplus \mathbf{a}_{k-1}) \oplus \mathbf{n}_k$$

Where:

- $\mathbf{s}_k$ is the state vector at time k.
- $\mathbf{C}$ is the connectivity matrix that defines the relationship between states.
- $\mathbf{a}_{k-1}$ is the action vector applied at time $k-1$ affecting the state transition.
- $\mathbf{n}_k$ is the state transition noise vector that introduces randomness into the transition.
- $\oplus$ is the XOR operation used for combining the state vector and action vector as well as adding the noise vector.

The transition matrices, representing the probability of moving from any state to other states under different control inputs is defined as:

$$(M(\mathbf{a}))_{ij} = p^{\|\mathbf{s}_i - (\mathbf{C}\mathbf{s}_i \oplus \mathbf{a})\|_1} \cdot (1-p)^{4 - \|\mathbf{s}_i - (\mathbf{C}\mathbf{s}_i \oplus \mathbf{a})\|_1}$$

where $\|\mathbf{v}\|_1 = \sum_i |\mathbf{v}(i)|$ is the 1-norm, summing the absolute values of the elements of the vector $\mathbf{v}$.

The reward function $R(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ is defined as:

$$R(\mathbf{s}, \mathbf{a}, \mathbf{s}') = 5s'(1) + 5s'(2) + 5s'(3) + 5s'(4) - |\mathbf{a}|$$

where $|\mathbf{a}|$ sums the absolute value of the elements of the action vector $\mathbf{a}$. The activation of each gene contributes a reward of +5, and actions $\mathbf{a}_2$ to $\mathbf{a}_5$ incur a cost of -1.

Q-Learning

a.

a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a1	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a3	a2	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a2	a2	a1	a2	a1	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a1	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a1	a2	a1	a2	a1	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a1	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a1	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a2	a2

b.

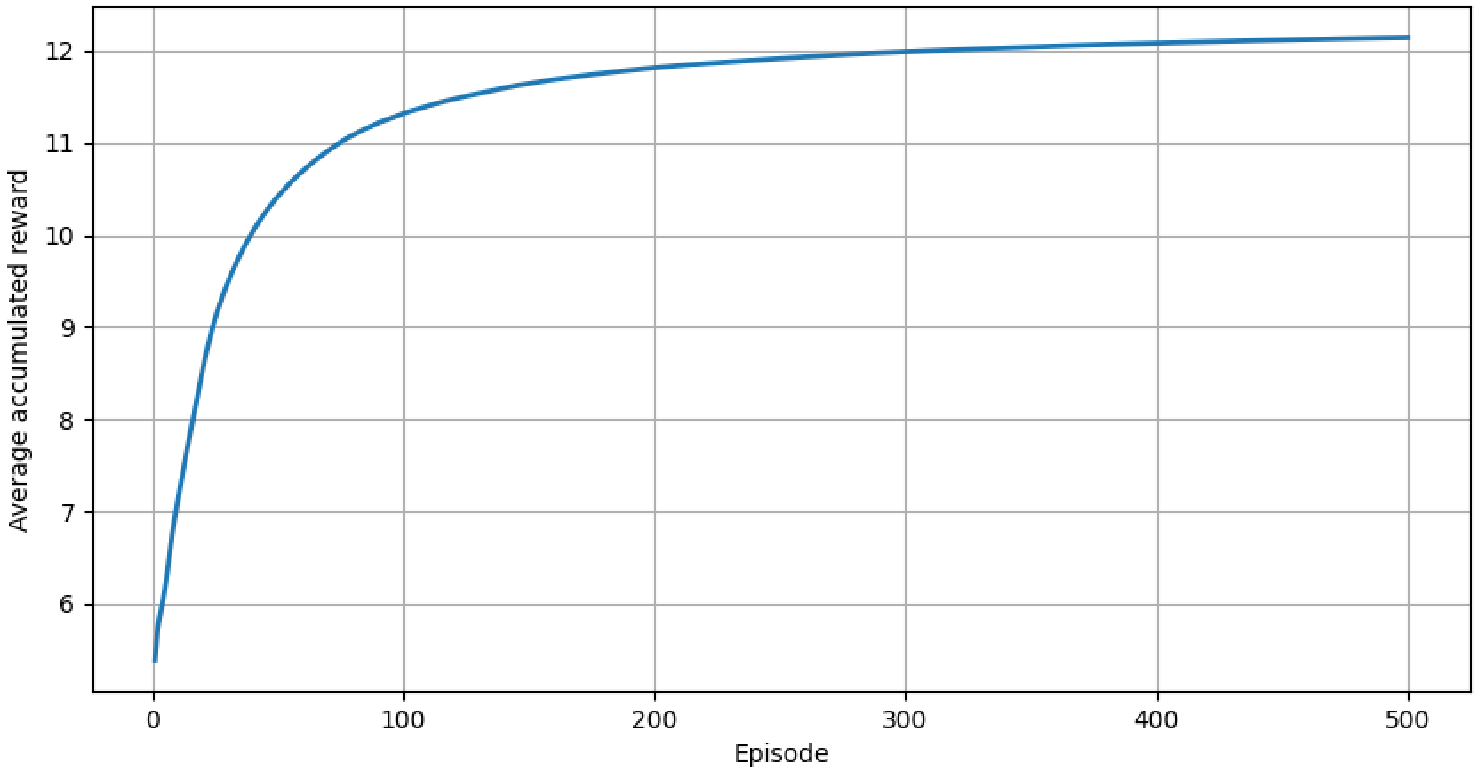

Figure 12: Average accumulated reward (in 10 independent runs) w.r.t episode number for Q-Learning

SARSA

a.

a2	a2	a2	a2	a2	a2	a2	a2	a3	a3	a1	a2	a1	a2	a2	a2
a2	a3	a2	a2	a2	a2	a2	a2	a1	a2	a2	a3	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a3	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a2	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a1	a1	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a1	a2	a2	a2
a2	a3	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a2	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a1	a2	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a1	a2	a1	a2

b.

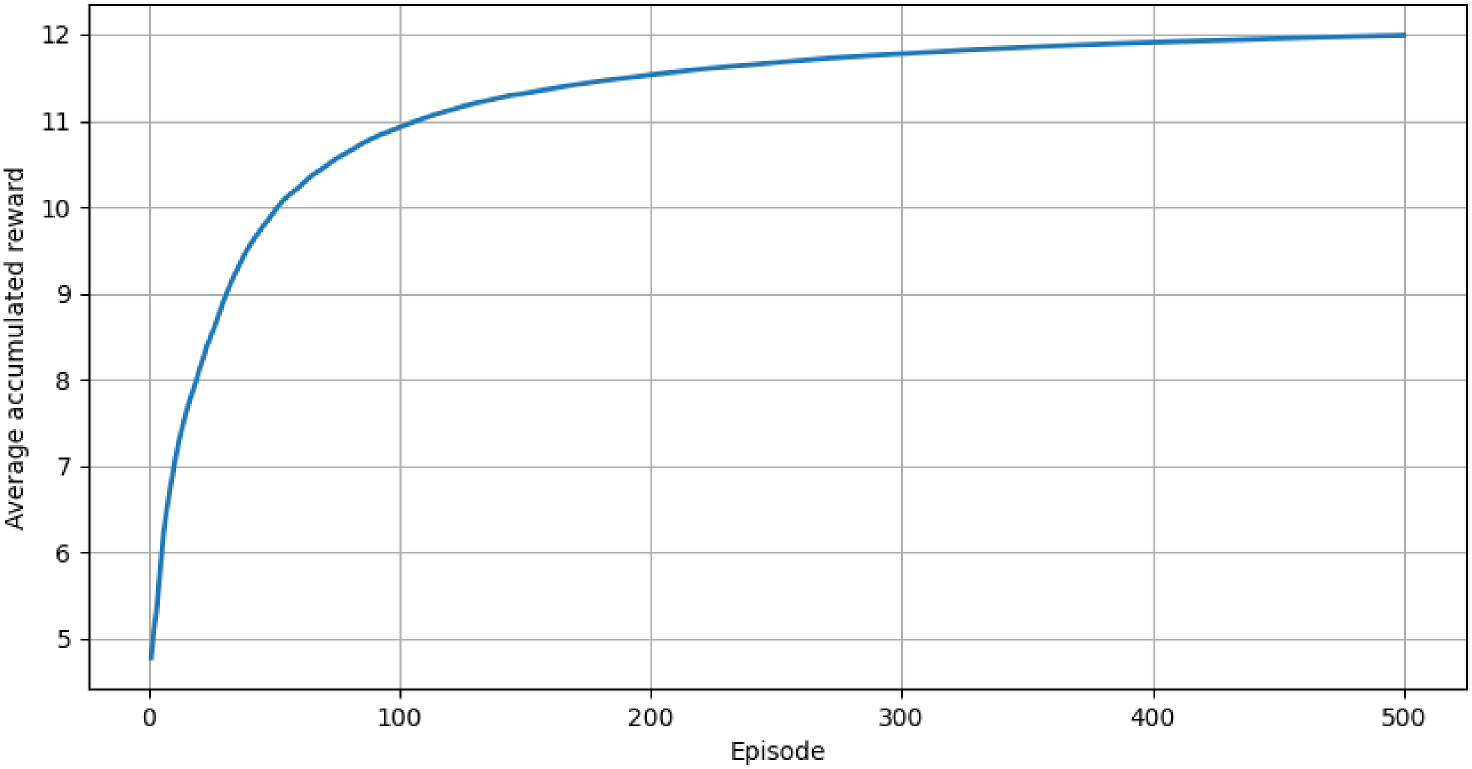

Figure 13: Average accumulated reward (in 10 independent runs) w.r.t episode number for SARSA

SARSA (λ)

a.

a2	a2	a1	a2	a2	a2	a2	a1	a3	a2	a2	a2	a1	a2	a2	a2
a3	a3	a3	a2	a1	a2	a2	a1	a3	a2	a2	a3	a1	a2	a1	a2
a2	a2	a3	a2	a2	a2	a1	a1	a1	a2	a3	a2	a1	a1	a1	a2
a2	a3	a2	a2	a2	a2	a2	a1	a1	a2	a3	a3	a1	a2	a1	a2
a3	a3	a2	a2	a1	a2	a2	a1	a1	a3	a2	a2	a2	a2	a1	a2
a2	a1	a3	a3	a2	a2	a2	a1	a1	a2	a2	a2	a1	a2	a1	a1
a3	a2	a2	a2	a1	a2	a1	a1	a1	a2	a3	a3	a1	a2	a1	a2
a3	a3	a2	a2	a2	a2	a2	a3	a1	a2	a2	a2	a1	a2	a2	a2
a2	a1	a2	a2	a2	a1	a1	a1	a1	a3	a3	a2	a3	a1	a2	a1
a2	a3	a2	a3	a2	a2	a2	a3	a1	a2	a3	a1	a1	a2	a1	a1

b.

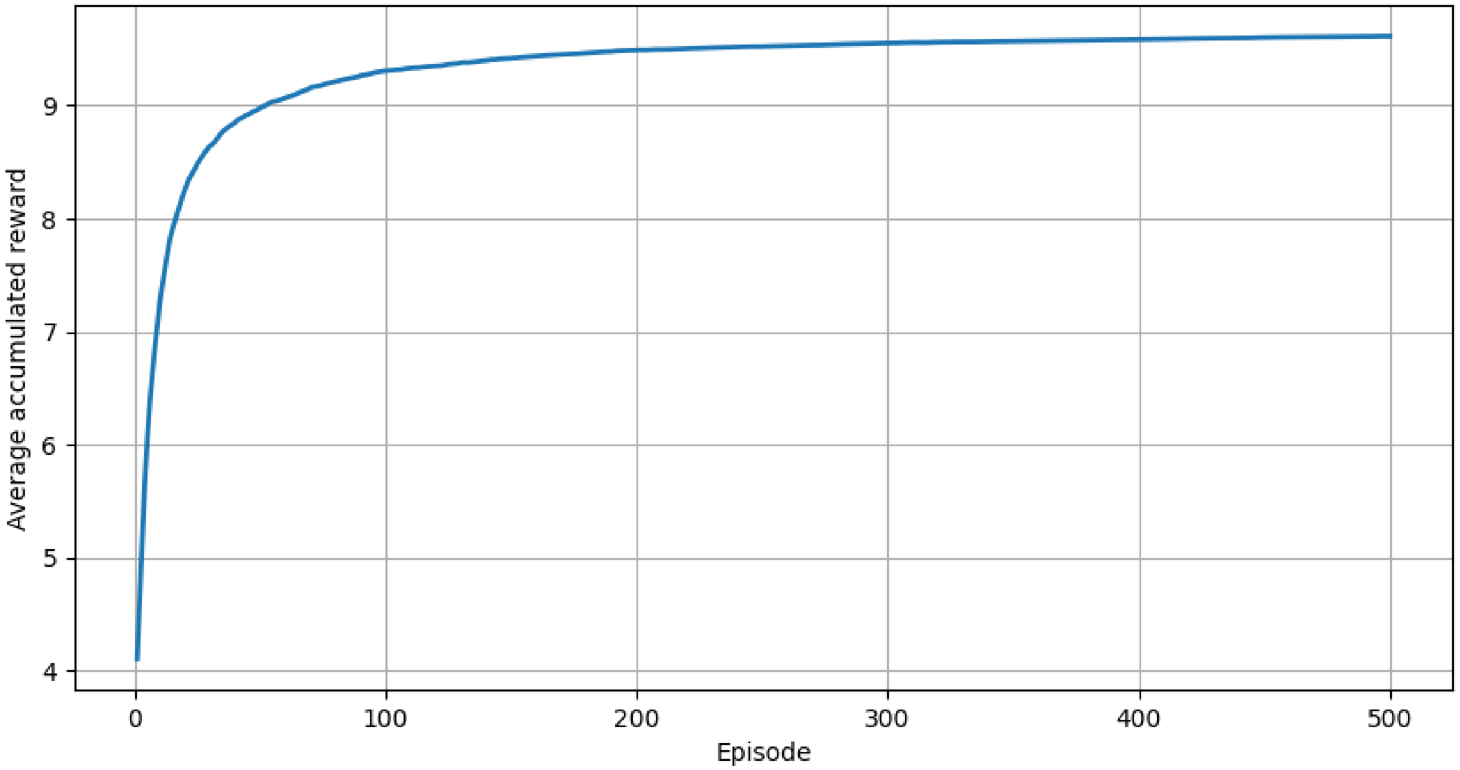

Figure 14: Average accumulated reward (in 10 independent runs) w.r.t episode number for SARSA (λ)

Actor-Critic

a.

a2	a2	a3	a3	a2	a2	a2	a2	a1	a2	a2	a2	a3	a1	a2	a2
a3	a2	a3	a2	a2	a2	a2	a2	a3	a2	a2	a2	a1	a2	a1	a2
a2	a2	a3	a2	a2	a2	a2	a2	a3	a3	a3	a2	a3	a2	a2	a2
a3	a3	a2	a2	a2	a2	a2	a2	a3	a3	a1	a3	a1	a2	a1	a2
a2	a3	a3	a3	a2	a2	a2	a2	a2	a2	a3	a2	a3	a2	a2	a2
a2	a2	a3	a2	a2	a2	a2	a2	a1	a1	a3	a2	a3	a1	a2	a2
a2	a2	a2	a2	a2	a2	a2	a2	a1	a3	a1	a2	a3	a2	a1	a2
a2	a3	a2	a2	a2	a2	a2	a2	a3	a3	a2	a2	a3	a2	a2	a2
a3	a2	a2	a2	a2	a2	a2	a2	a3	a3	a3	a2	a1	a2	a2	a2
a2	a3	a2	a2	a2	a2	a2	a2	a2	a2	a3	a2	a1	a2	a2	a2

b.

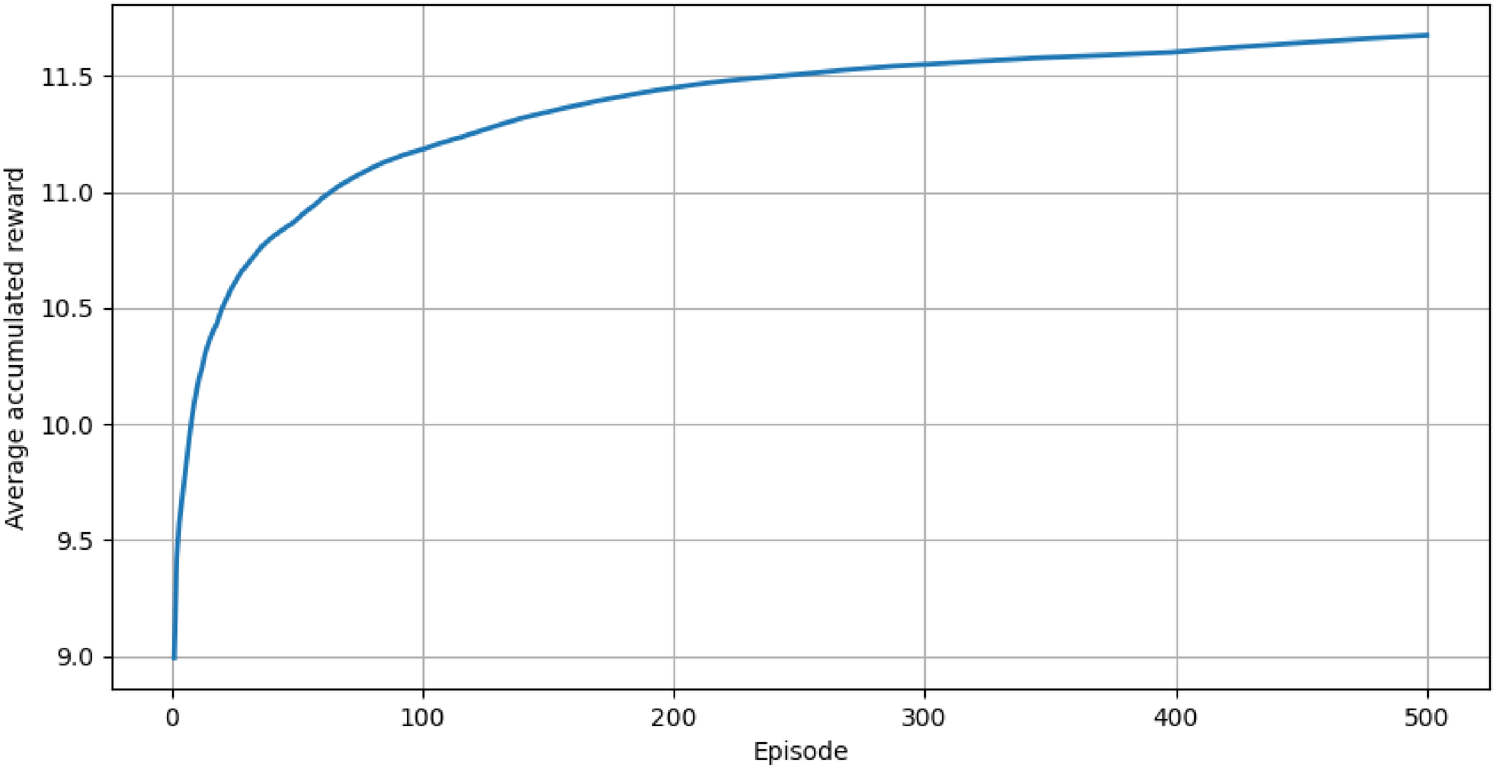

Figure 15: Average accumulated reward (in 10 independent runs) w.r.t episode number for Actor-Critic

c.

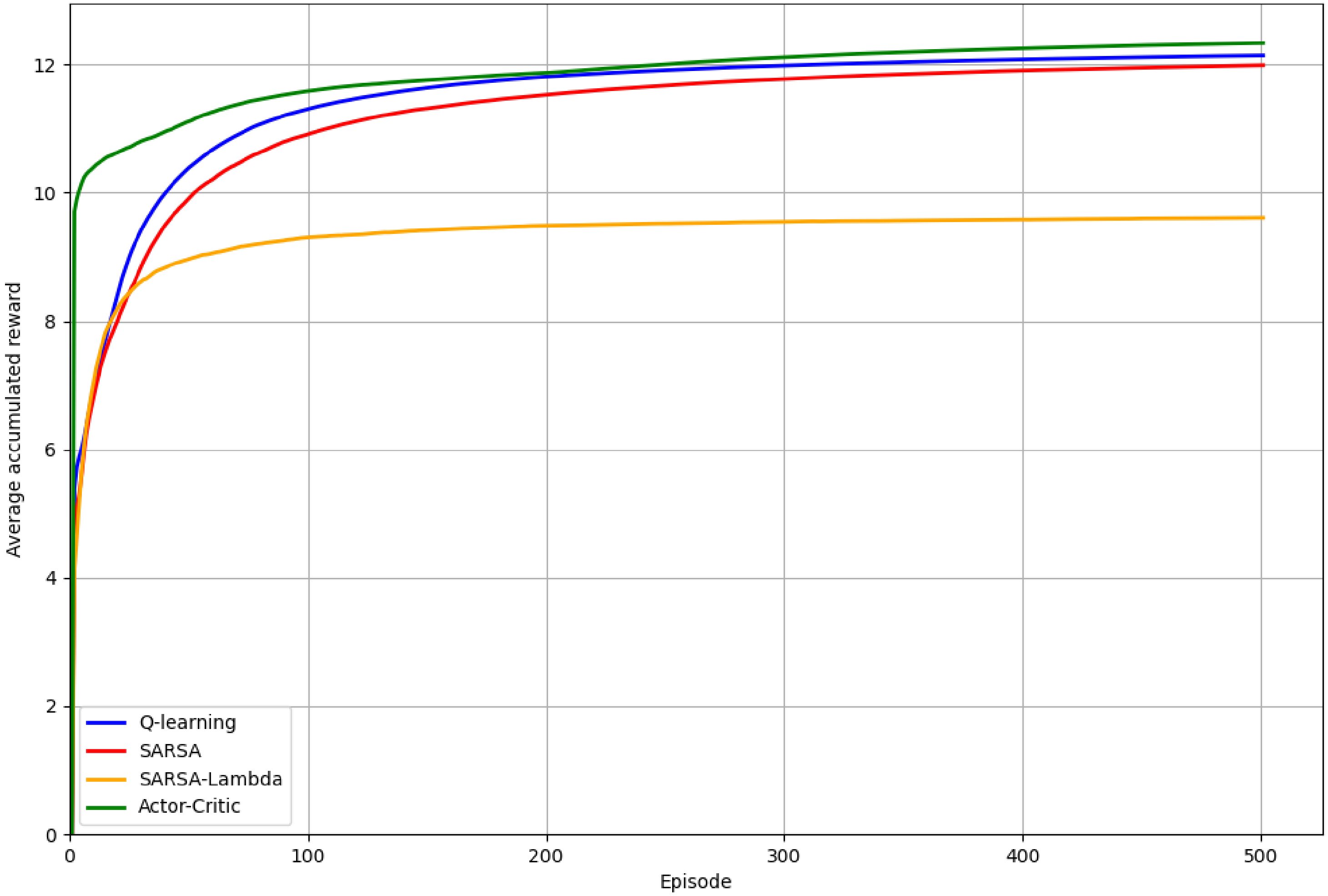

Figure 16: Average accumulated reward (in 10 independent runs) w.r.t episode number for all algorithms

Analysis

Q-learning is an off-policy algorithm that estimates the value of the optimal policy by learning the action-value function.This estimation process is carried out regardless of the agent's current actions, which enables Q-learning to evaluate the potential of actions that have not been taken based on observed outcomes allowing the Q-learning algorithm to discover the optimal policy, even in environments high stochasticity environments.In the initial learning phase (Episode 0-100), the Q-learning curve displays a steep ascent, suggesting that the algorithm is rapidly identifying actions that lead to high rewards and making significant progress towards the optimal policy. This is indicative of Q-learning's preference of exploitation of gathered information in comparison to exploration to make substantial policy improvements without being constrained by the policy defined exploratory actions.During the mid-learning phase (Episode 100-300), the Q-learning curve's slope decreases,highlighting a reduction in the rate of improvement as the algorithm balances exploration (new actions to evaluate) and exploitation (using the known best actions) to stabilize converging policy.During the late learning phase (Episode 300-500), the Q-learning curve plateauing indicates that the Q-learning algorithm has converged to a stable policy that is close to the optimal policy for the stochastic environment and reward function.

From the graph, it can be observed that during the initial learning phase (Episodes 0-100), the SARSA curve exhibits a quick ascent. This highlights the on-policy nature of SARSA, which updates its policy based on the actions taken, including exploratory actions. The conservative rise of the SARSA curve suggests a cautious learning strategy of on-policy algorithms, which directly integrate actual exploratory actions into policy updates.In the mid-learning phase (Episodes 100-300), the SARSA algorithm continues to make progress with a slowing rate of improvement, indicative of the algorithm stabilizing and converging toward a policy that balances exploration with exploitation.The leveling off of the SARSA curve below the Q-learning curve suggests that the SARSA algorithms on-policy approach leads to a more conservative policy, possibly due to the reinforcement of sub optimal actions taken during exploration.The SARSA algorithm accumulates fewer rewards compared to a more exploitative strategy such as Q-learning, as the algorithm utilizes exploration for policy updates.During the late learning phase (Episode 300-500), the SARSA curve plateaus, indicating the algorithm has converged to a reliable stable policy, but not an optimal policy as the accumulated reward is not maximized.

From the graph, it can be observed that during the initial learning phase (Episodes 0-100),the SARSA((λ)) curve demonstrates a moderate ascent and then plateaus, indicating a relatively slower learning process due to the SARSA((λ)) algorithms on-policy nature which promotes balance of exploitation and exploration.During the mid-learning phase (Episode 100-300),the SARSA((λ)) curve indicates a relatively conservative update strategy due to the leveling off a lower accumulated reward suggesting a limitation or ineffectiveness of the eligibility traces in providing optimal action information from past experiences.In the late learning phase (Episodes 300-500),the SARSA((λ)) plateaus suggesting the algorithm's sensitivity to the stochasticity of the environment, leading to less optimal policy.

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 100-300)the Actor-Critic algorithm continues to improve which indicates that the actor is refining policy based on new information provided by the critic adjusting policy in response to long-term rewards.In the late learning phase (Episodes 300-500), the Actor-Critic curve indicates convergence towards the optimal policy with highest average accumulated reward, suggesting that it has determined the most optimal policy for the stochastic environment. The actor-critic algorithm is able to perform well in a complex and stochastic environment as it can make incremental changes and refine policy based on the actor critic feedback loop.