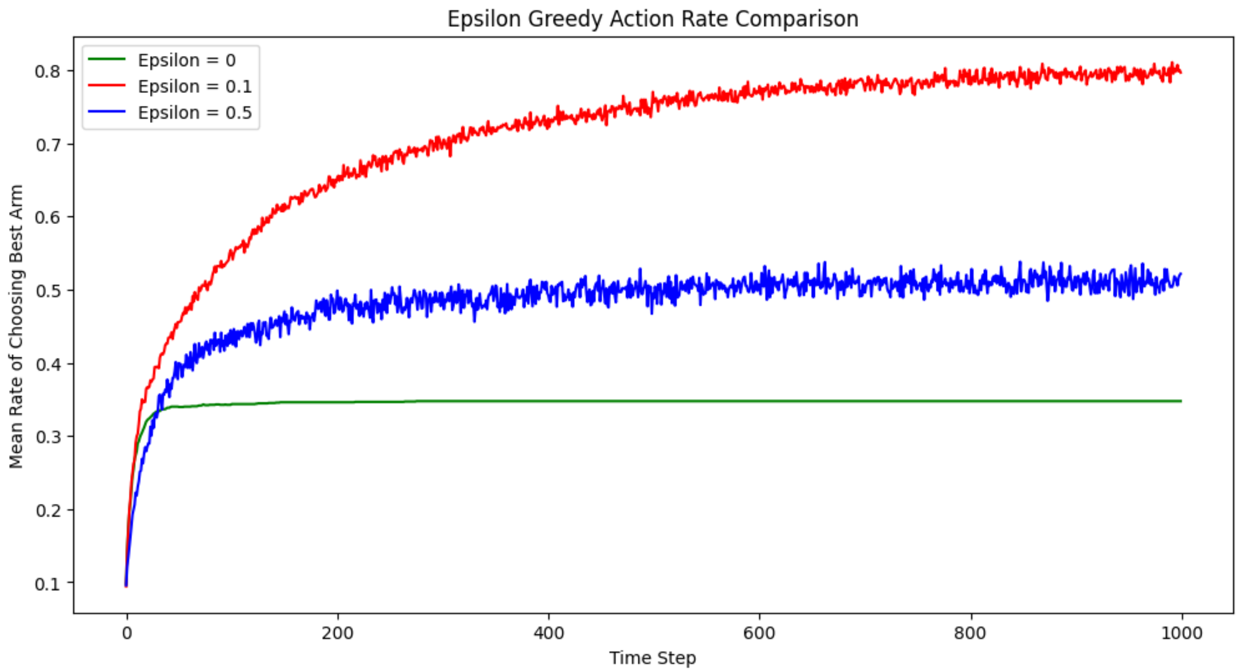
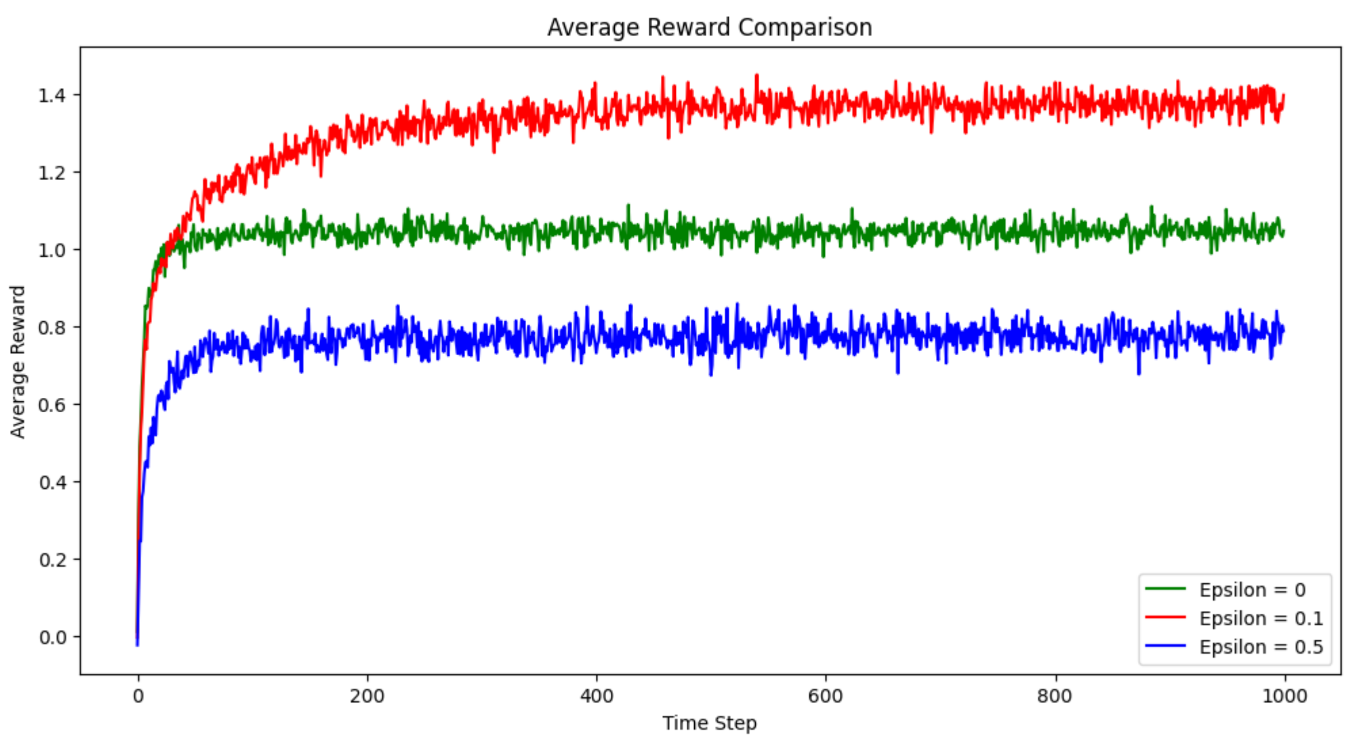
Results
Bandit Problem

(1) Epsilon Greedy Method Plots:

Plot 1: This plot compares the **average rewards** obtained by the **epsilon-greedy algorithm over time steps for different values of epsilon.**



Plot 2: This plot compares the **rate** at which the **epsilon-greedy algorithm chooses the best arm over time steps for different values of epsilon.**

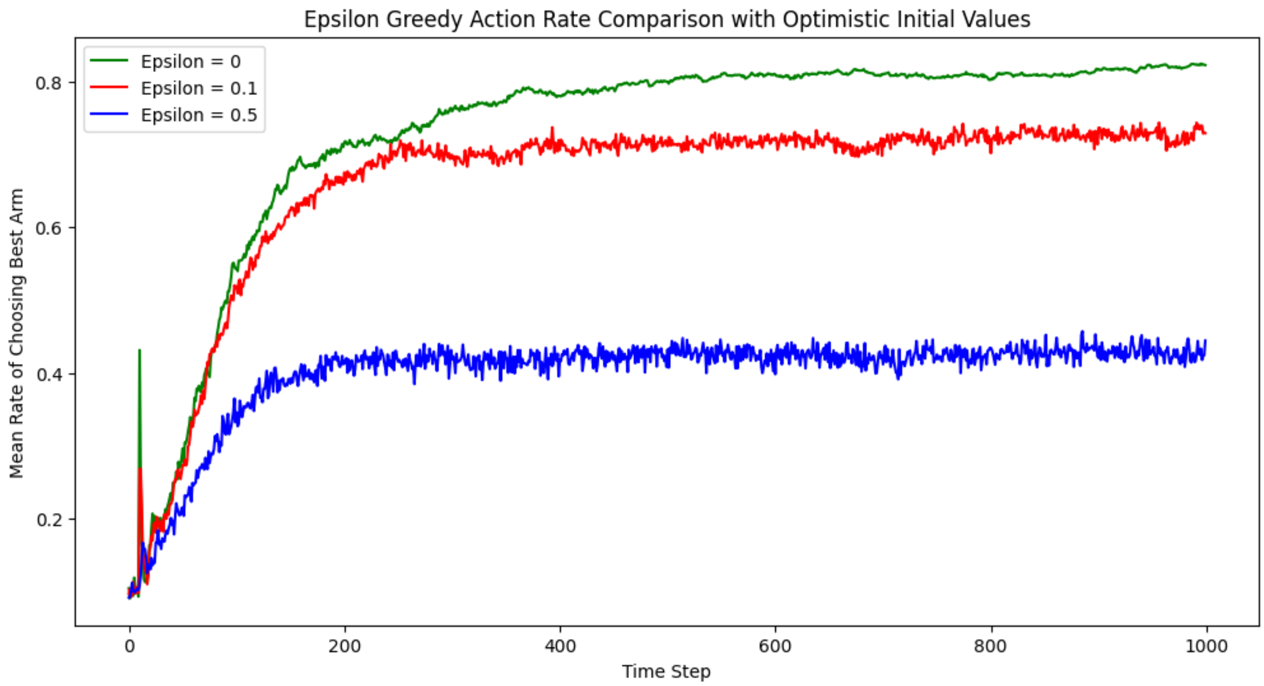


Conclusion :

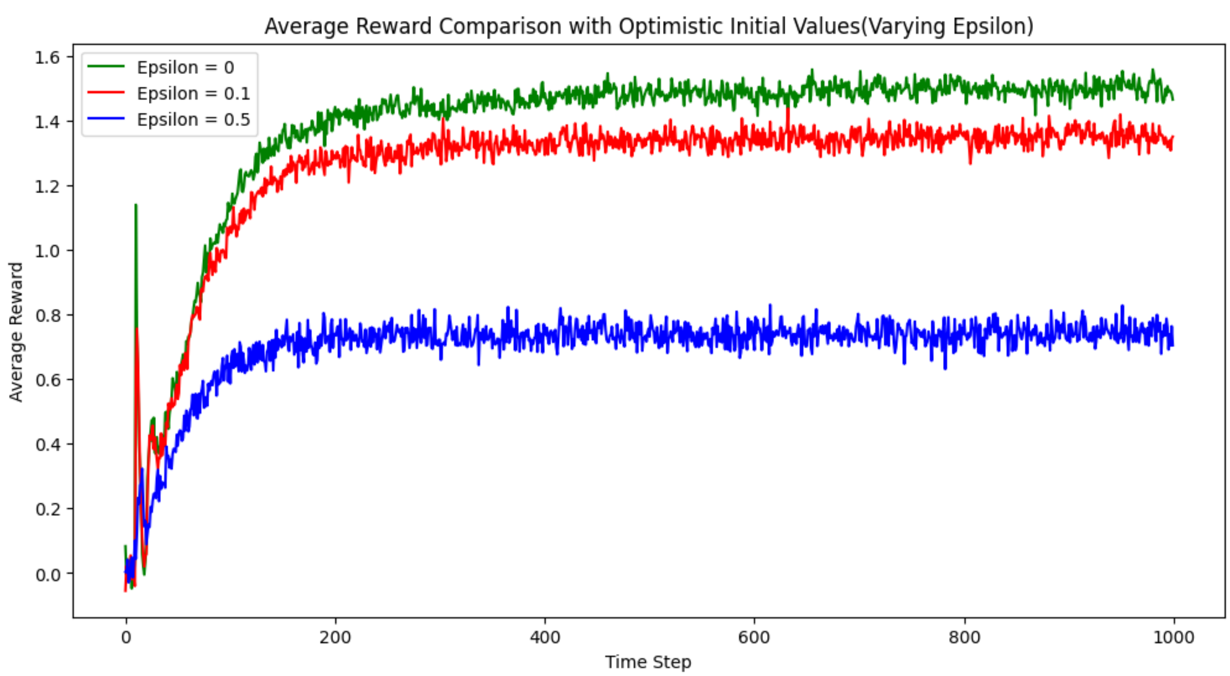
1. As you increase the value of epsilon (ε), the algorithm becomes more explorative.
2. Higher epsilon values result in a **higher percentage of exploration**, which means the algorithm explores suboptimal actions more frequently.
3. Consequently, the percentage of times the optimal action is selected decreases as epsilon increases.
4. However, increasing epsilon can lead to higher average rewards during the early stages of learning due to increased exploration.

(2) Optimal Initial Value :

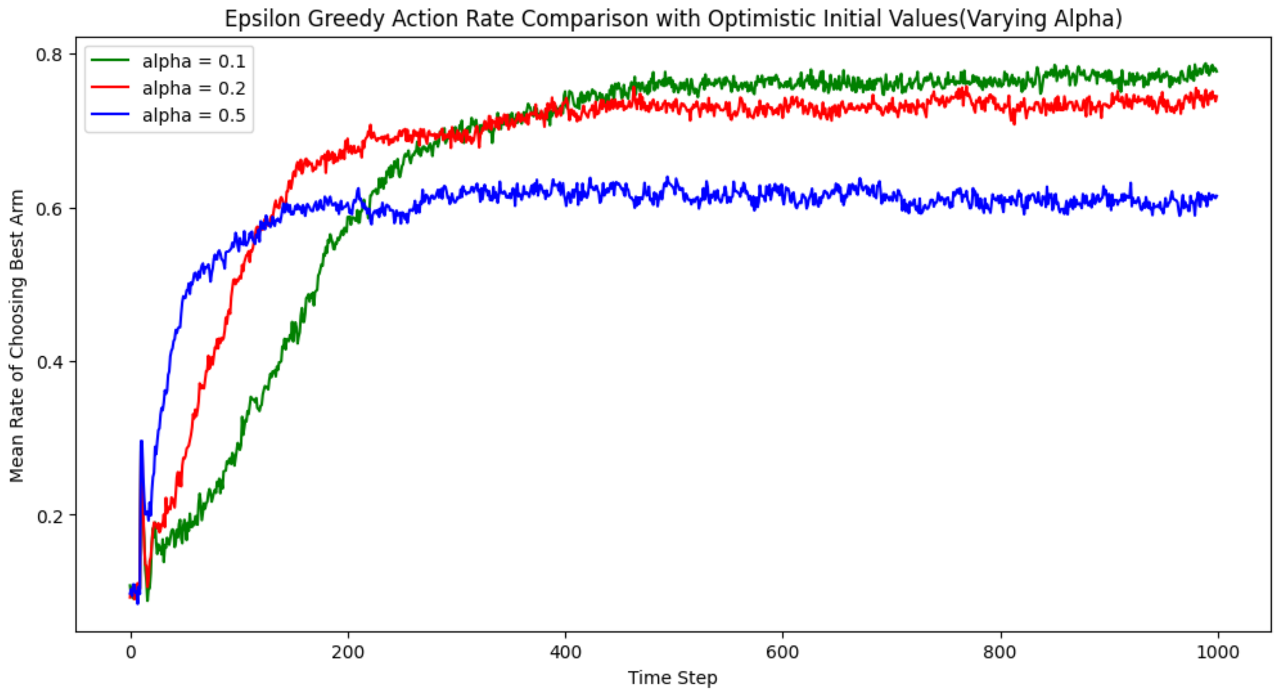
Plot 3: This plot compares the **average rewards** obtained by the **epsilon-greedy algorithm with optimistic initial values** over time steps for different values of epsilon.



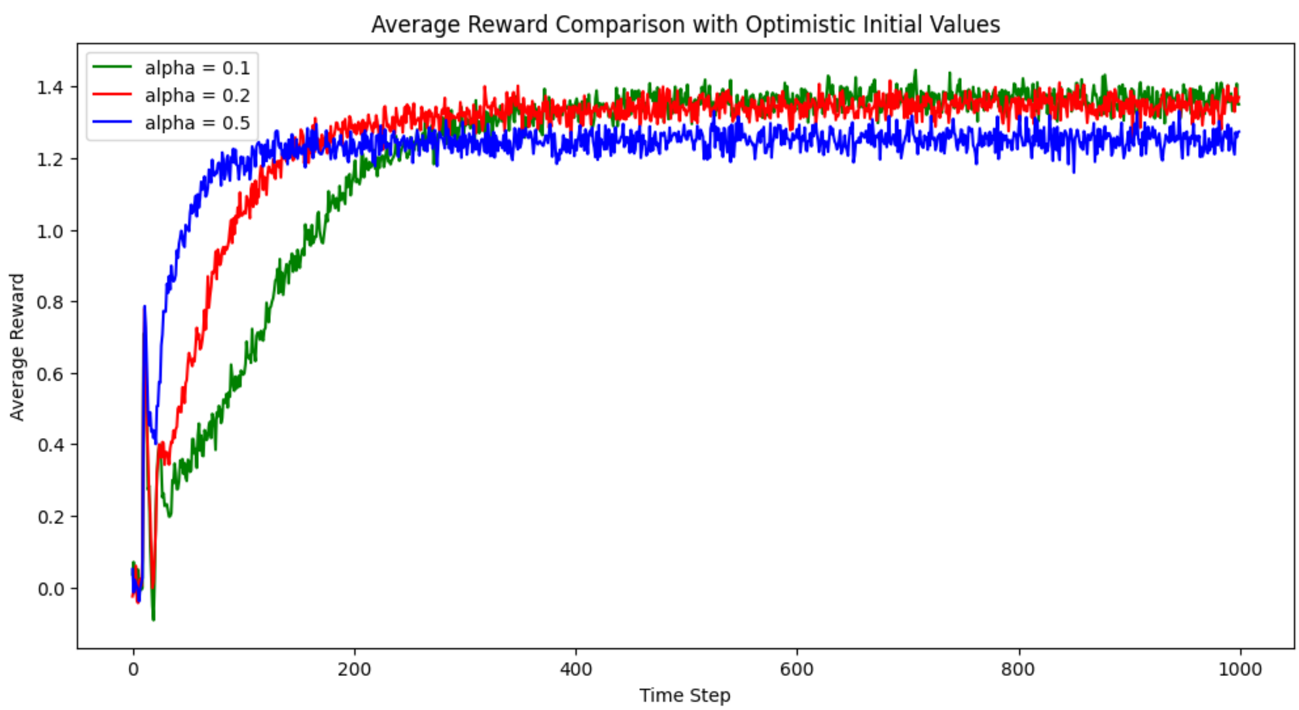
Plot 4: This plot compares the **rate** at which the **epsilon-greedy algorithm with optimistic initial values** chooses the best arm over time steps for **different values of epsilon.**



Plot 5: This plot compares the rate at which the epsilon-greedy algorithm with optimistic initial values chooses the best arm over time steps for different values of the step size parameter alpha.



Plot 6: This plot compares the **average rewards** obtained by the epsilon-greedy algorithm with optimistic initial values over time steps for different values of the step size parameter alpha.

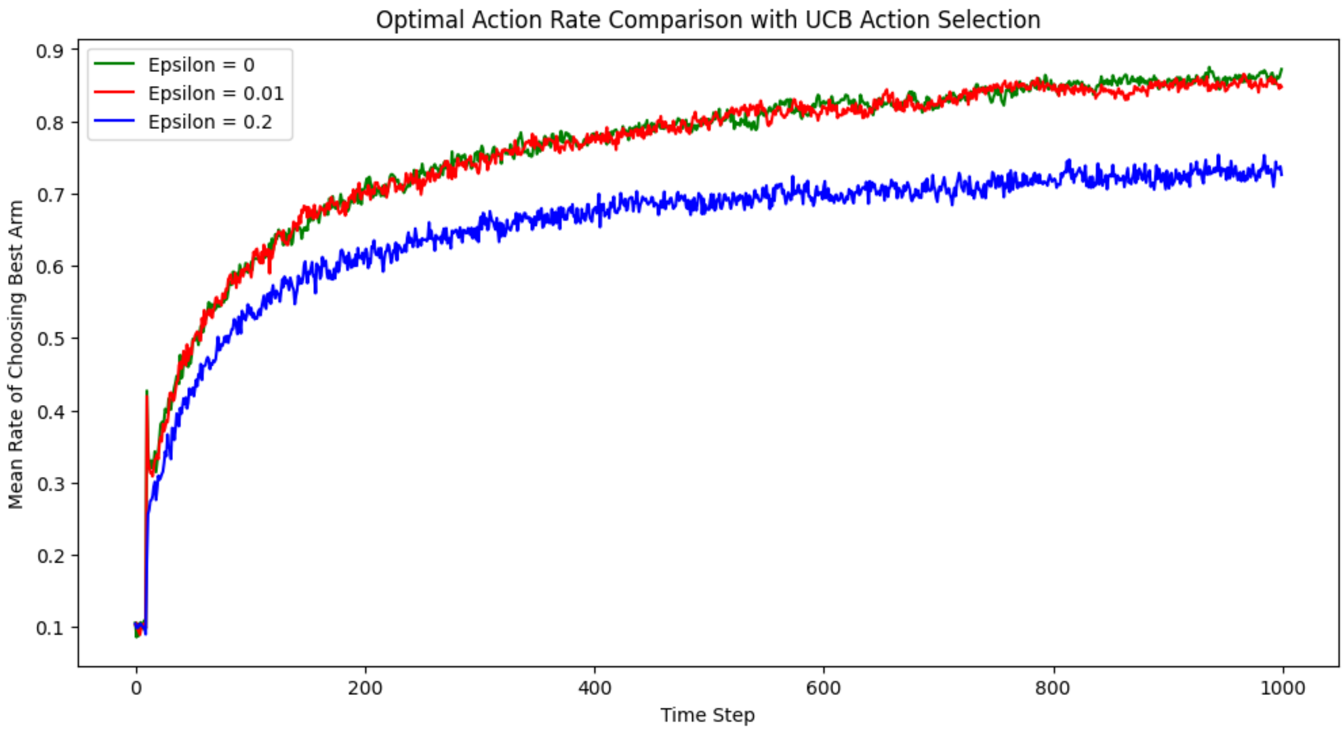


Conclusion :

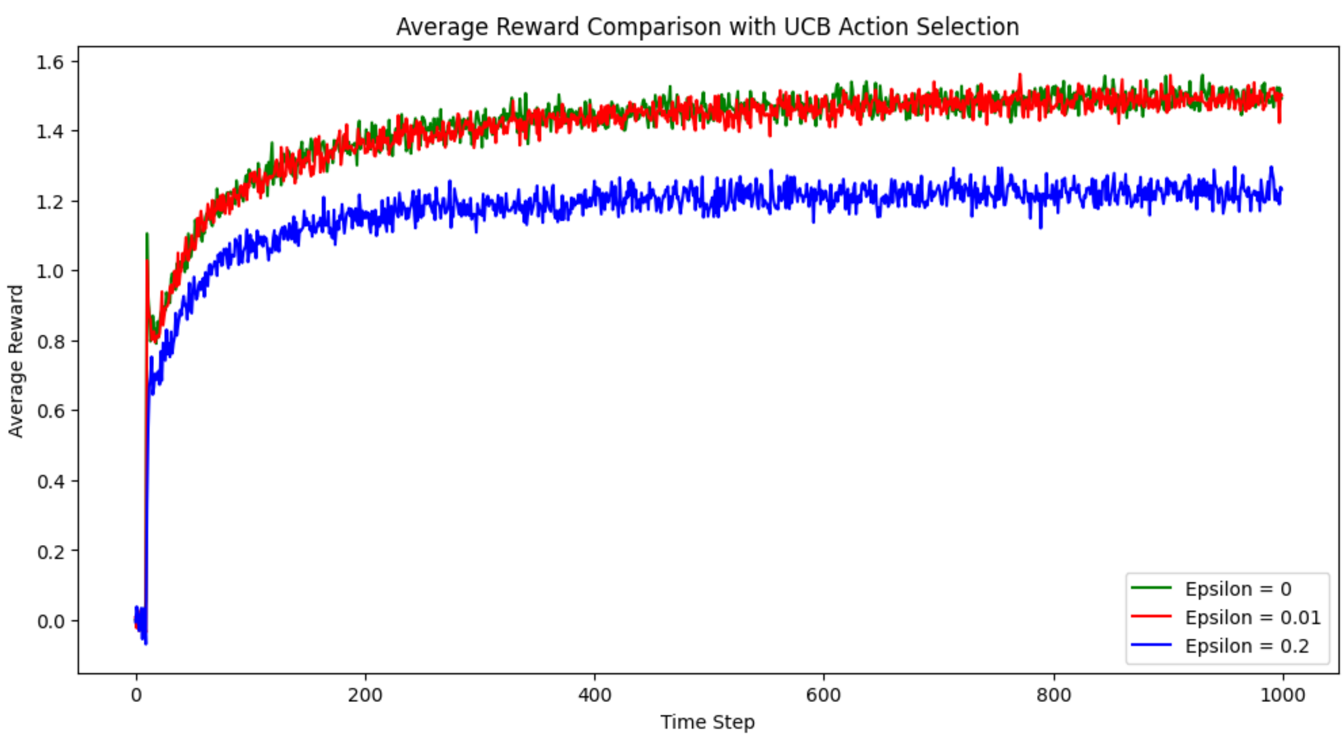
1. Providing optimistic initial values for action estimates encourages more exploration.
2. With optimistic initial values, the algorithm starts with a positive bias for all actions, encouraging it to explore.
3. This often results in a higher percentage of times the optimal action is selected during the initial episodes.
4. Over time, as the algorithm learns more about the actual action values, the percentage may stabilize.

(3) UCB :

Plot 7: This plot compares the **rate** at which the **epsilon-greedy algorithm with UCB Action selection values** chooses the best arm over time steps for **different values of epsilon.**



Plot 8: This plot compares the **average rewards** obtained by the **epsilon-greedy algorithm with UCB action selection** over time steps for different values of epsilon

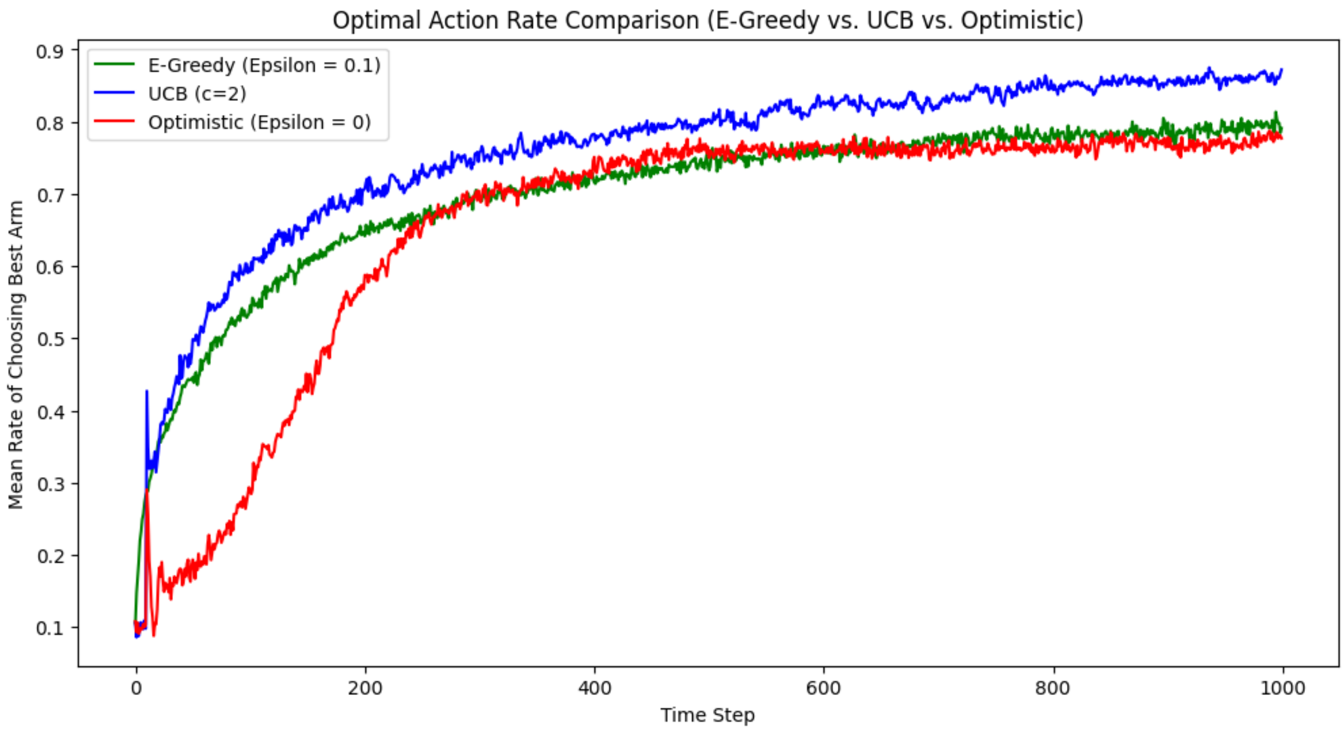


Conclusion :

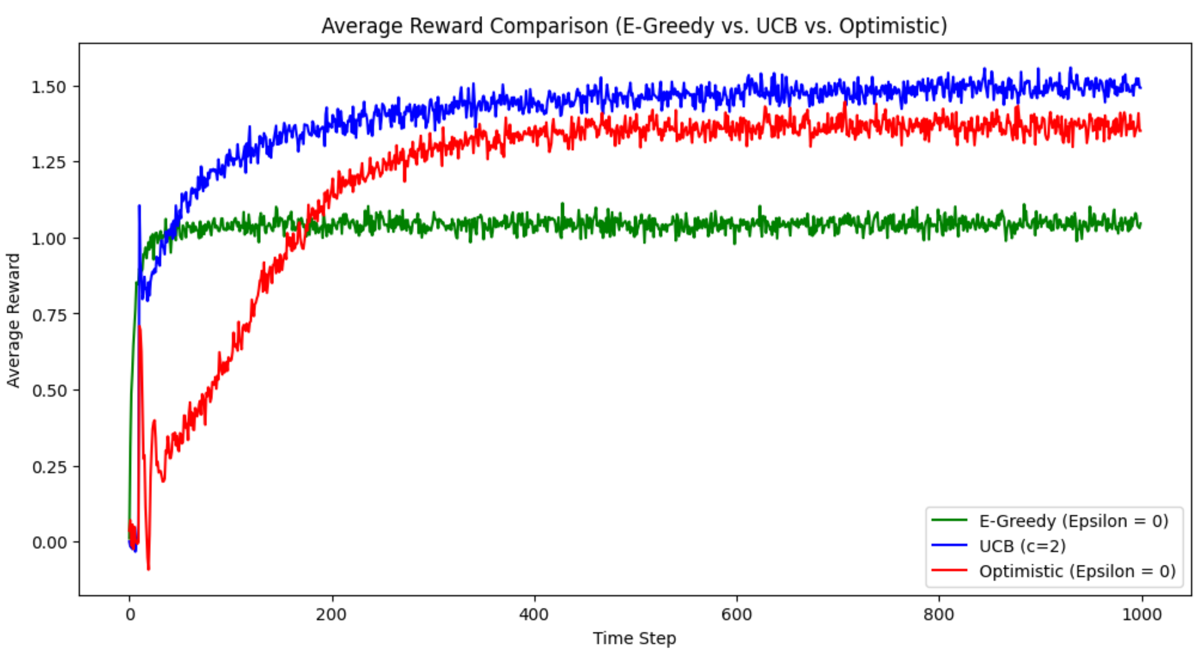
1. These graphs compare epsilon-greedy algorithms with UCB action selection.
2. Green (Epsilon = 0): UCB tends to perform better than epsilon-greedy with no exploration due to the inherent exploration from UCB.
3. Red (Epsilon = 0.01): With a small epsilon, UCB combines exploration and exploitation effectively.
4. Blue (Epsilon = 0.2): Higher epsilon values in UCB lead to more exploration.

(4) Comparison:

Plot 9: This plot compares the rate at which the epsilon-greedy, UCB, and optimistic algorithms choose the best arm over time steps.



Plot 10: This plot compares the average rewards obtained by the epsilon-greedy, UCB, and optimistic algorithms over time steps



Conclusion :

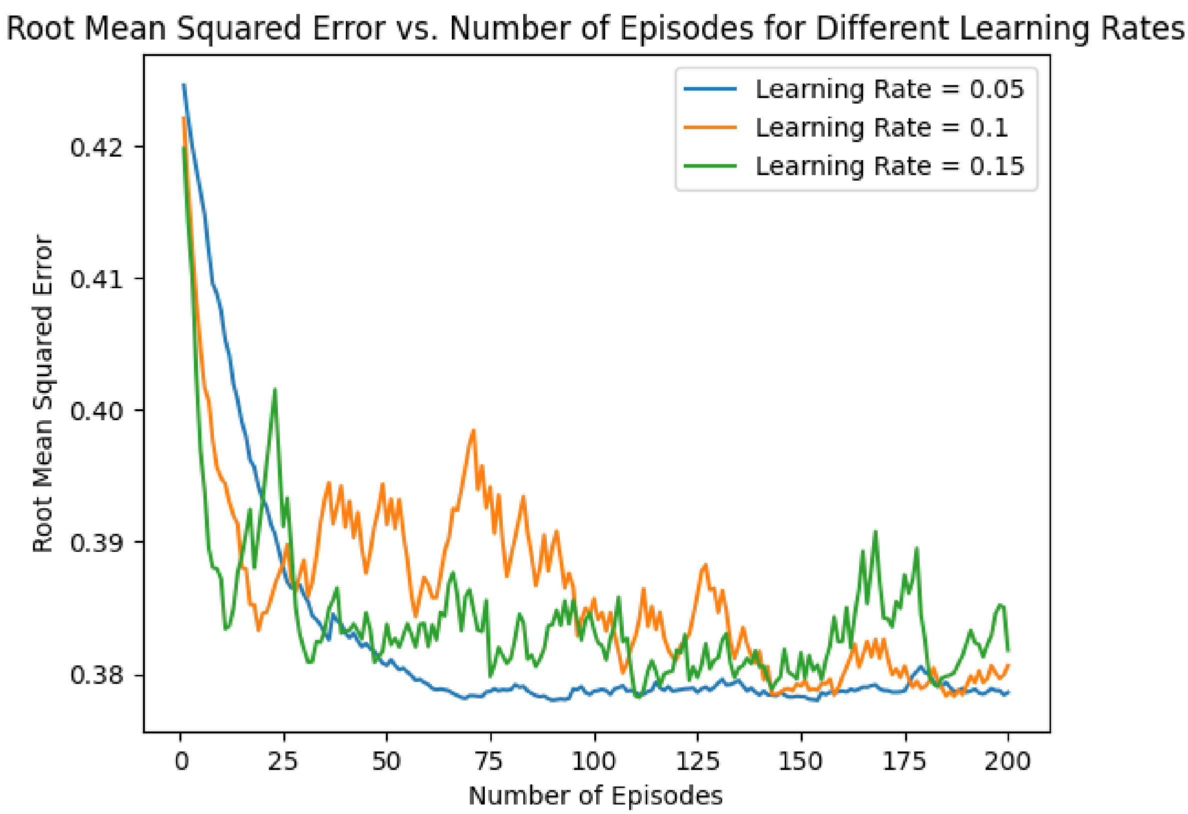
1. The last set of graphs compares the optimal action rates and average rewards between epsilon-greedy, UCB, and optimistic initial values.
2. Epsilon-Greedy (Epsilon = 0.1): Strikes a balance between exploration and exploitation, performing well over time.
3. UCB (c = 2): UCB also performs well, providing better exploration than epsilon-greedy without sacrificing exploitation.
4. Optimistic (Epsilon = 0): Optimistic initial values help exploration but may perform poorly if not balanced.

Overall Conclusion:

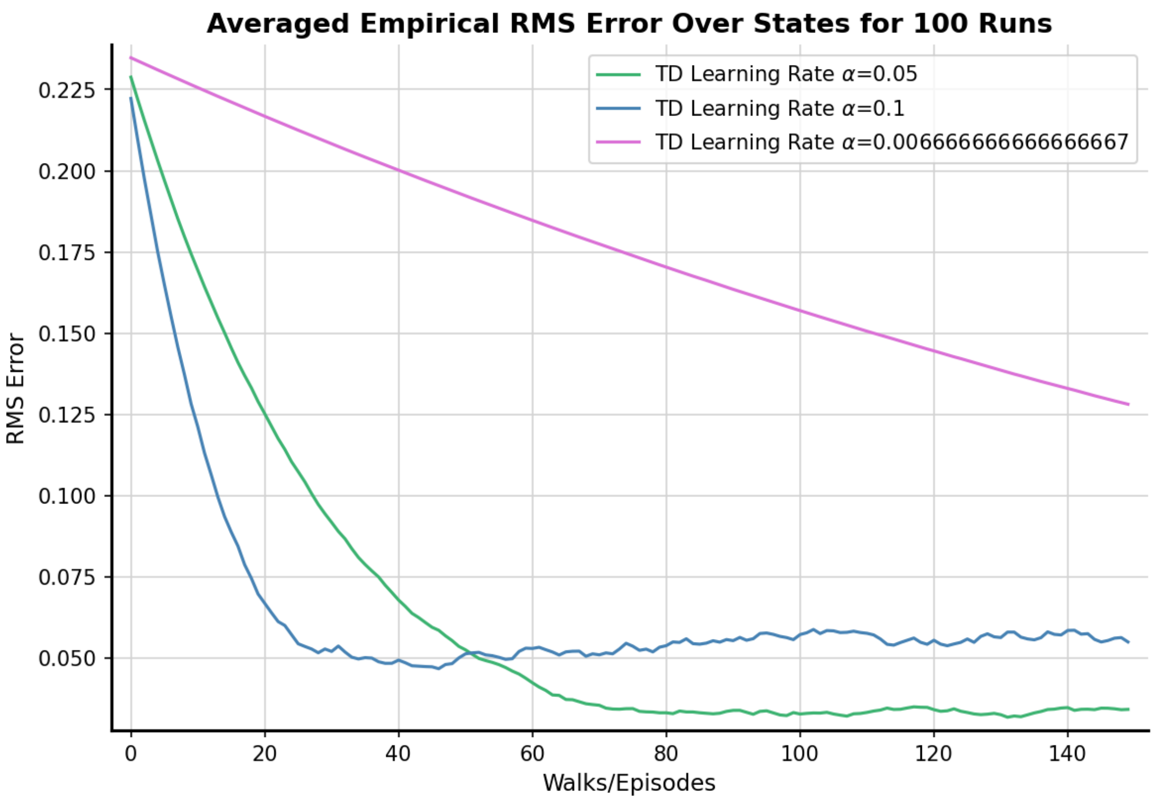
1. The choice of epsilon significantly impacts exploration vs. exploitation.
2. Optimistic initial values can enhance early exploration.
3. Alpha values affect the learning rate, influencing how quickly the algorithm adapts to new information.
4. UCB combines exploration and exploitation effectively.
5. The optimal action rate and average reward depend on the specific parameters chosen.

MRP Problem

Plot 11: This plot shows how the root mean squared error (RMSE) between estimated state values and true state values changes as the number of episodes increases for different learning rates in a TD evaluation.



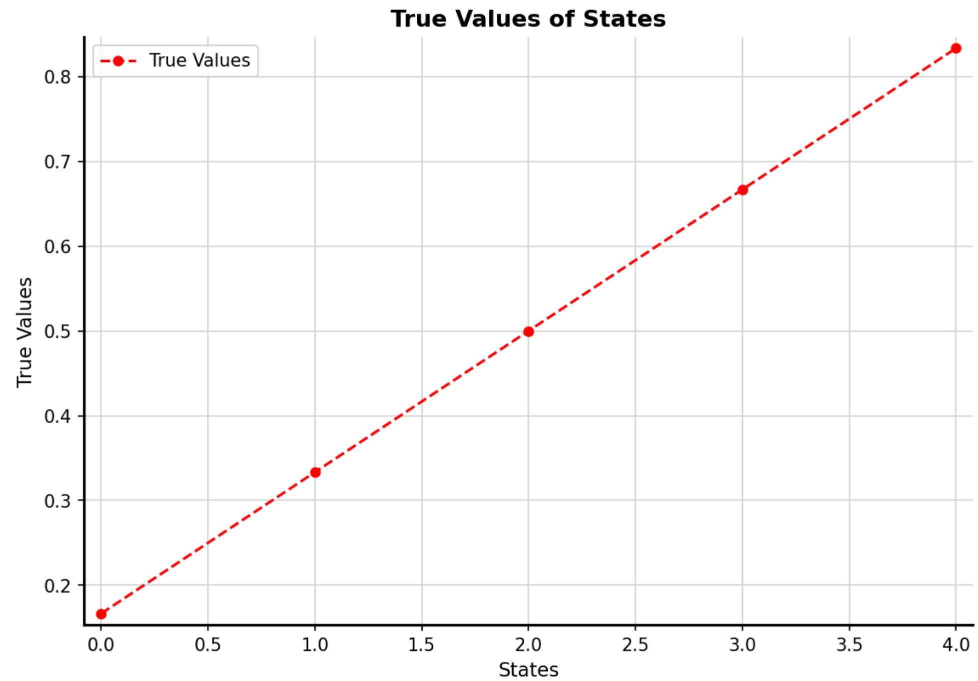
Plot 12: This plot shows how the root mean squared error (RMSE) between estimated state values and true state values changes as the number of episodes increases for different learning rates in a TD evaluation.



Observation :

1. This graph compares the TD(0) algorithm's performance with different learning rates (alpha values).
2. Observation: Smaller alpha values (0.05) result in slower convergence, while larger values (0.15) converge faster but with more variance. The intermediate value (0.1) balances convergence and stability.

Plot 13: This plot displays the true values of states in the Markov Random Process (MRP).



Conclusion :

1. TD(0) with various alpha values exhibits different convergence rates and stability.
2. Smaller alpha values converge more slowly but result in smoother learning curves.
3. Larger alpha values lead to quicker convergence but may exhibit more variance.
4. As the number of episodes increases, RMSE tends to decrease, indicating improved approximation of true state values.
5. The root-mean-squared errors generally converge to zero, especially for intermediate alpha values.
6. When alpha = 1/n (sample average update rule), the root-mean-squared errors will converge to zero. This is because the sample average update rule ensures that each observed reward contributes equally to the estimate, leading to better convergence.