Forest Management & Taxi Markov Decision Process

Yumna Rizvi
CS7641 – Assignment 4
Yrizvi3@gatech.edu

Abstract: This study focuses on two distinct Markov Decision Process (MDP) problems: the Taxi and Forest Management problems. The Taxi-v3 problem, a grid-world scenario with 500 states, presents a complex environment for evaluating algorithm performance in route optimization and decision-making. Meanwhile, the Forest Management problem, analyzed in both small (20 states) and large (500 states) versions, offers a contrast in problem scales and demonstrates decision-making under uncertainty and long-term planning. These problems collectively provide a comprehensive platform for examining the efficacy of Value Iteration, Policy Iteration, and Q-Learning algorithms, particularly in varying state space sizes. **Keywords**: *Markov's Decision Process, Reinforcement Learning, Value Iteration, Q-Learning, Policy Iteration*

1. Introduction to the problems

Taxi Problem

The Taxi problem, a classic in the field of reinforcement learning, involves a taxi navigating a 500-state grid to pick up and drop off passengers. This setup, serving as a benchmark in the field, allows for an in-depth analysis of how algorithms like Value Iteration (VI) and Policy Iteration (PI) scale with complexity. It also acts as a testbed for Q-learning, demonstrating the algorithm's capabilities in a large, complex state space. This problem facilitates a direct comparison with the Forest Management problem, highlighting the impact of state space size on algorithm performance.

Forest Management Problem

The Forest Management problem, embodying sequential decision-making with long-term impacts, is explored in both small and large state spaces. This dual approach enables a detailed analysis of algorithm behavior and performance across different scales, underlining the challenges posed by uncertainty and long-term strategy formulation. Tackling this problem in two different sizes allows a direct comparison with the Taxi problem.

Q-Learning Algorithm

An epsilon greedy Q-learning, chosen for its suitability in stochastic and discrete environments, offers a balanced approach to exploration and exploitation. Its adaptability and performance in both the Taxi and Forest Management problems underscore its effectiveness in diverse scenarios, particularly in larger state spaces. These chosen problems and the Q-learning algorithm together provide diversity in a comprehensive exploration of MDPs, showcasing algorithmic strengths and decision-making challenges in varied contexts, particularly on how state space size impacts the performance and effectiveness of different algorithms.

Q-learning, being a model-free algorithm, may initially struggle with exploration in the vast state space of the Taxi problem. However, it's expected to eventually converge to an optimal or near-optimal policy, balancing exploration and exploitation effectively. For the small Forest Management problem- due to the smaller state space, Q-learning is likely to learn the optimal policy relatively quickly, with less exploration required compared to the larger state spaces. However, in the larger version of the problem, Q-learning may exhibit slower convergence due to the need for extensive exploration. The algorithm's performance will heavily depend on the balance between exploration and exploitation and the setting of learning parameters.

Value Iteration

The expectation for VI on the Taxi problem is that it will efficiently converge to an optimal policy due to the deterministic nature of the Taxi problem's environment. However due to state space, this may lead to increased computation time. Meanwhile, for the small Forest Management problem, VI is expected to rapidly find an optimal policy, given the problem's relatively straightforward decision-making process and fewer states. For the larger version of the Forest Management problem, VI might face challenges in convergence speed due to the increased complexity and the need for more iterations to evaluate the long-term consequences of decisions.

Policy Iteration

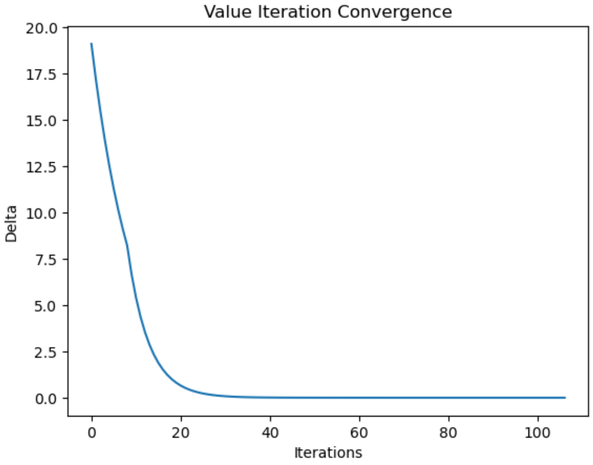
While expected to be efficient, the Taxi problem may require significant computational resources for policy evaluation and improvement phases due to the state space. For the smaller Forest Management Problem, PI is expected to converge quickly to an optimal policy, as the smaller state space allows for rapid policy evaluations and improvements.

The larger state space in this variant may lead to slower convergence for PI, as each iteration involves evaluating a policy over many states, which could be computationally intensive.

2. Taxi

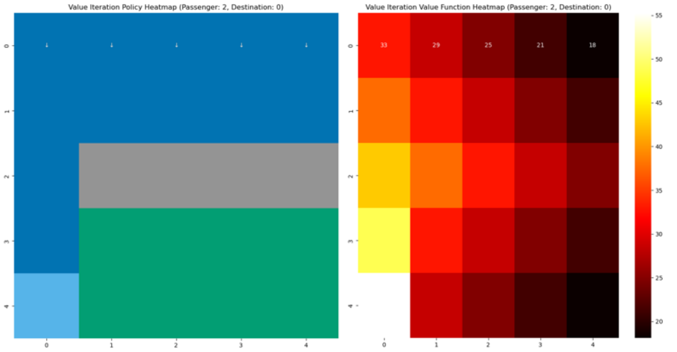
The Taxi-v3 problem in reinforcement learning provides a challenging and dynamic environment with a discrete action space of 6 possible actions and a state space consisting of 500 distinct states. The discrete nature of both the action and the state space is conducive to applying VI and PI, which rely on the environment's Markov property.

2.1 Value Iteration



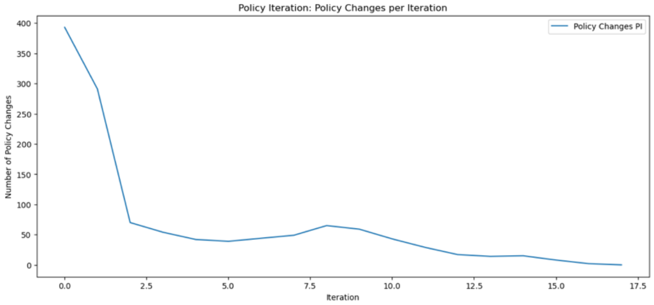
The VI algorithm was initialized with a zero-value function and iteratively updated using the Bellman optimality equation. The stopping criterion was set with a small threshold (theta), ensuring that the algorithm converged to a sufficiently optimal policy. The convergence of VI was observed at iteration 106, indicated by the delta plot which measures the maximum change in the value function across all states from one iteration to the next. The steep decline in the delta curve

illustrates a rapid improvement in the value estimates, eventually flattening out, indicating convergence.
Repeated runs of the algorithm consistently converged at iteration 106, showing the deterministic nature of the algorithm's convergence given the environment and the initialization parameters.

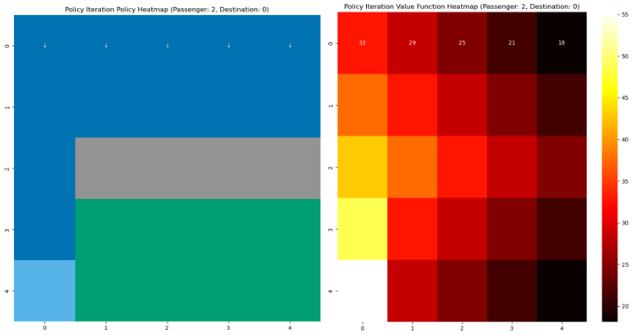


The policy extracted from the VI process demonstrates a deterministic mapping from states to actions, which is expected in a Markov Decision Process with discrete states and actions. The experiment with the Taxi-v3 environment indicates that VI is a robust algorithm capable of effectively converging to an optimal policy within a reasonable number of iterations, even in a relatively large state space. This is consistent with the theoretical properties of VI, which guarantees convergence to an optimal policy if all states are updated in each iteration. For future studies or applications, it would be beneficial to further explore the effect of different discount factors (gamma) on the convergence rate and the quality of the resulting policy. Moreover, further tuning the threshold (theta) might provide insights into the trade-off between computational efficiency and the optimality of the policy.

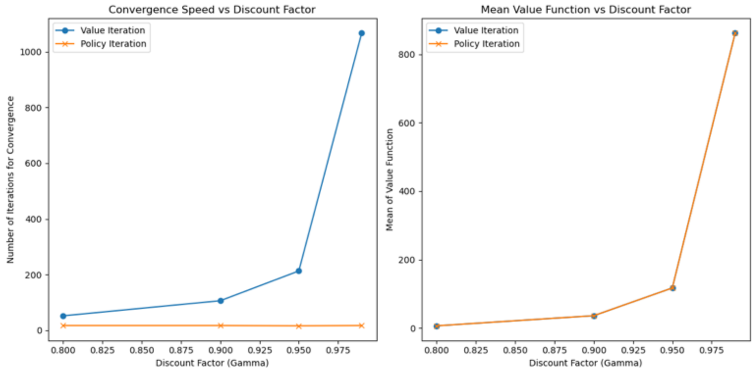
2.2 Policy Iteration



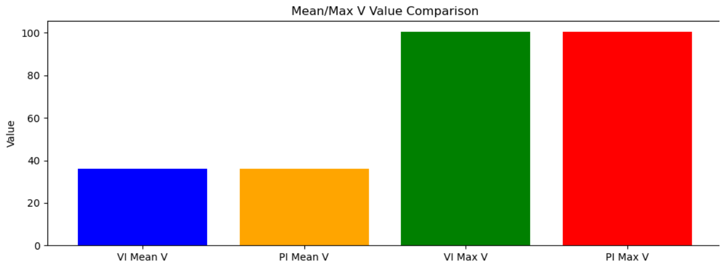
PI consists of two main steps: policy evaluation and policy improvement, which are iterated until the policy converges. The randomized initial policy for PI adds variability to the convergence process, contrasting with the deterministic nature of VI. PI showed convergence within a range of 15 to 30 iterations, with most runs converging by the 17th iteration. This variability in convergence times for PI, as opposed to the consistent convergence of VI at 106 iterations, may be attributed to the stochastic initialization of policies in PI and the two-step iterative process it employs. Despite this variability, PI generally converged faster than VI, reflecting PI's potential for efficiency in certain problem structures.



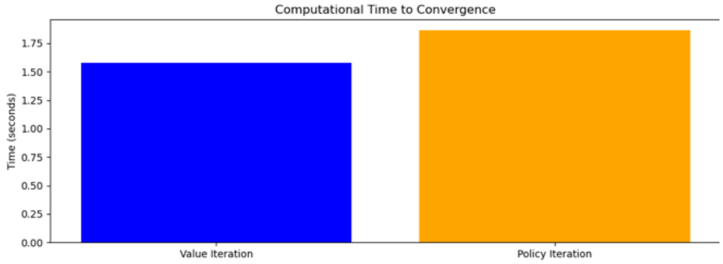
The policies generated by PI were consistent with the deterministic policies required for optimality in MDPs. This consistency suggests that PI, even with its initial random policy, can effectively navigate the decision-making space of Taxi to find optimal routes. The analysis reaffirms the utility of PI in solving discrete MDPs and provides a practical comparison of their performance in a controlled environment.
The policy heatmaps for both VI and PI show a clear, interpretable policy across the state space. The heatmaps illustrate which actions the algorithms have determined to be optimal in each state, with certain patterns emerging that indicate structured strategies for picking up and dropping off passengers. The value function heatmaps reflect the expected cumulative reward from each state, under the optimal policy. The gradients in the heatmaps show the relative value of the states, with brighter colors indicating higher values. These visualizations show which states are most valuable and may be critical decision points within the environment.
While both algorithms converged to effective policies, the visual differences in the heatmaps highlight unique aspects of each algorithm's approach to learning. The VI heatmaps show a certain level of uniformity, reflecting the consistent approach of VI in updating values across all states. In contrast, the PI heatmaps may exhibit more variation due to the initial random policy and the iterative process of policy evaluation and improvement.



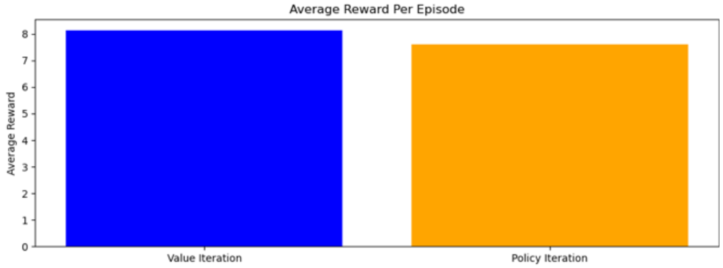
VI seems to require more iterations than PI as the discount factor increases, suggesting that PI may be more efficient in this setting when considering the discount factor's impact on convergence speed. Both algorithms increase as the discount factor approaches 1. However, PI's mean value function increases at a higher rate than VI's, indicating that PI may be more sensitive to changes in the discount factor within the Taxi problem's dynamics. When looking at the Mean/Max V Value comparison VI has a higher mean and maximum V value, which could suggest that it finds a policy with better overall value across states, or it could also indicate that VI overestimates the value function compared to PI.



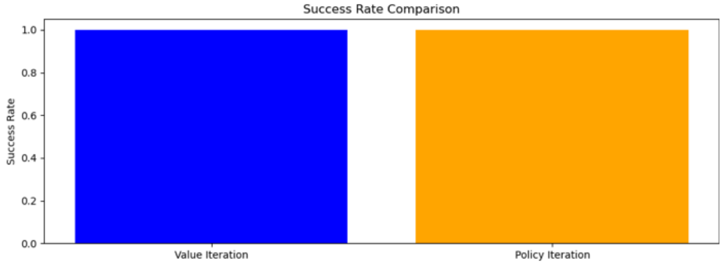
VI takes less time to converge compared to PI. This might be counterintuitive given PI's usually faster convergence in theory



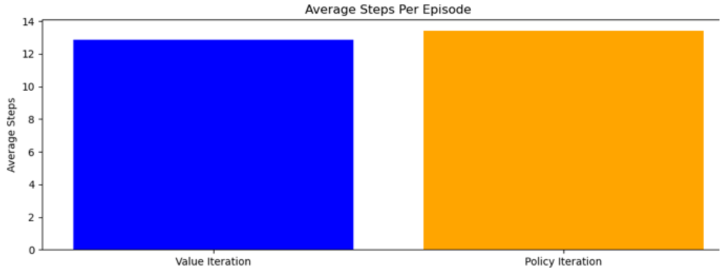
The average reward per episode is higher for PI than VI. This metric is crucial because it reflects the practical effectiveness of the learned policy when interacting with the environment.



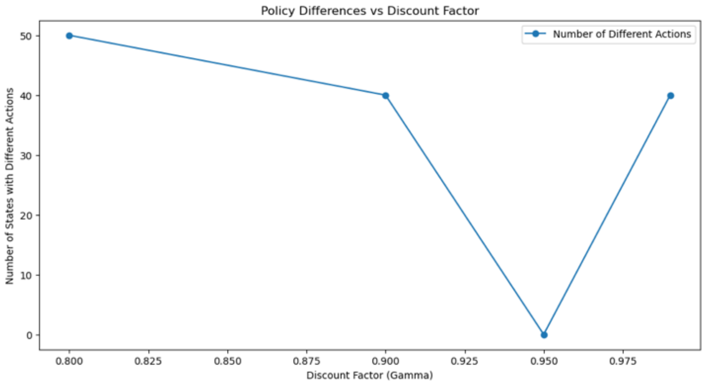
Both VI and PI show high success rates, but PI appears slightly lower. This could be due to PI converging on a policy that is less robust to certain states or episodes in the Taxi problem.

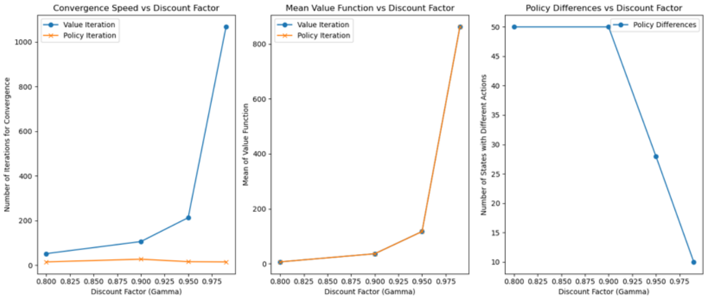


VI results in fewer average steps per episode compared to PI, suggesting that the VI policy is more efficient at solving the Taxi problem within fewer moves.

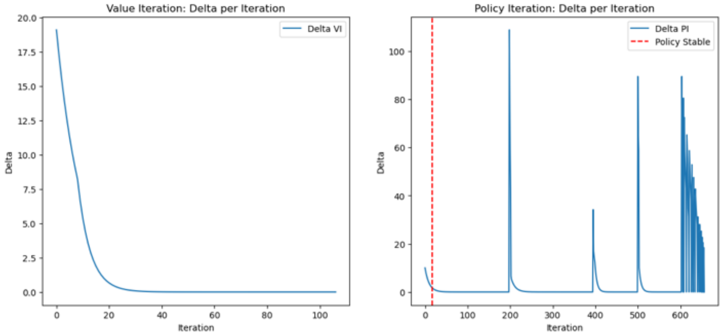


When looking at the policy differences vs discount factor, A significant drop in differences at a discount factor of around 0.9 shows a convergence of policy agreement between the algorithms, possibly indicating a threshold where both algorithms begin to find similar optimal policies.





The plots below illustrate VI is converging quickly to a stable value function. However, the plot for PI Delta per Iteration shows spikes, which might indicate that PI is making significant policy changes even at later iterations, a pattern not typically expected and which might suggest instability or a complex value landscape in the Taxi problem.

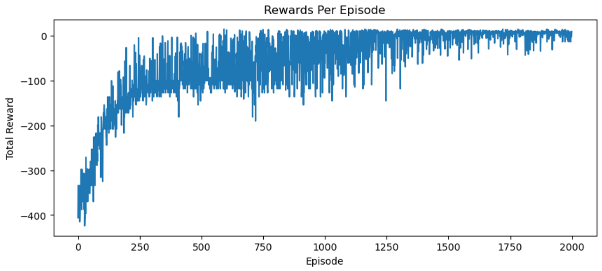


The faster convergence of PI, as opposed to VI's consistent but slower convergence, demonstrates the differences in how these algorithms approach policy optimization. The experiment has demonstrated that VI and PI can find optimal policies within the Taxi-v3 environment. PI, with its potential for faster convergence, may offer computational advantages over VI, especially in environments where the value function can be evaluated efficiently. The outcomes suggest that PI, despite the randomness in initial policies, can efficiently solve the Taxi problem, often faster than VI. Further investigation into the conditions leading to the variability in PI's convergence time could provide insights for optimization. Additionally, experimenting with different gamma values for the discount factor and varying the convergence threshold could fine-tune the balance between computational speed and policy optimality.

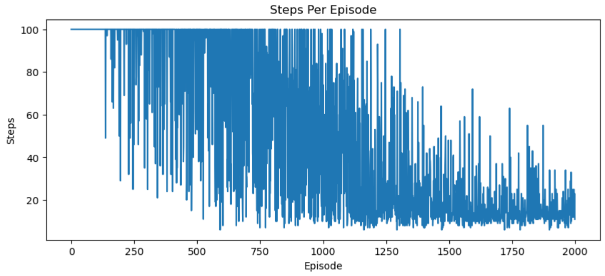
2.3 Q-Learning

The results for Q-Learning in the Taxi problem suggest that there is an improvement in rewards and a reduction in the number of steps as the algorithm progresses, which is a sign of the learning process. The consistent low epsilon value suggests that the learning is heavily skewed towards exploitation.

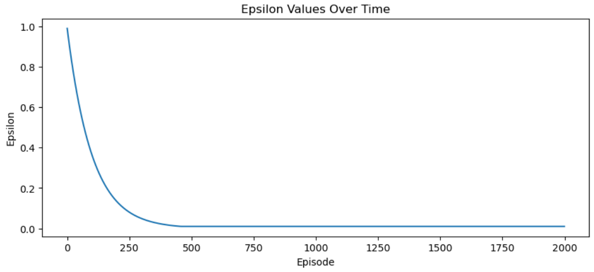
Episode	Reward	Steps	Epsilon
0	-62	73	0.01
100	-29	49	0.01
200	11	9	0.01
300	10	10	0.01
400	14	6	0.01
500	-10	21	0.01
600	9	11	0.01
700	-6	26	0.01
800	13	7	0.01
900	5	15	0.01



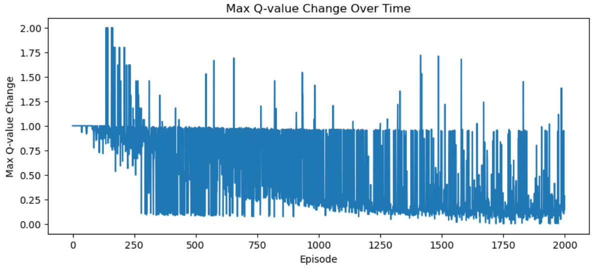
For rewards per episode, initially, the agent incurs high negative rewards, which gradually increase to consistently positive values. This progression indicates that the Q-learning agent is learning to complete the taxi task more efficiently over episodes.



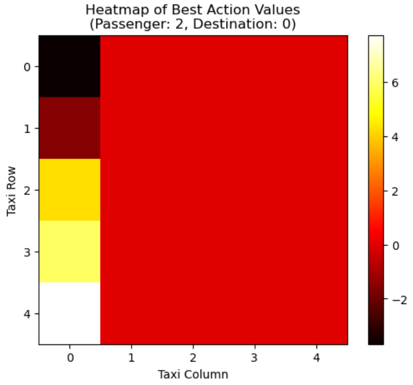
This plot shows learning efficiency as a clear downward trend is visible, meaning the agent is learning to solve the taxi problem in fewer steps.



When looking at the epsilon parameter's decay over time, there is a rapid decline and leveling off imply that the agent shifts from exploring the environment to exploiting the learned policy quickly.



The fluctuations in the max Q-value changes further indicate that the agent continues to learn and update its policy throughout the training process. The presence of spikes suggests significant learning events, possibly when the agent discovers better strategies.



The heatmap shows the Q action-value function for a specific state where the passenger is at location 2 and the destination is 0. The red areas have the highest Q-values, suggesting that these states/actions are more valuable for the given passenger and destination configuration.

Q-Learning shows a clear learning curve with improvement over episodes, while VI and PI typically converge to an optimal policy in one pass through the state space with multiple sweeps. Furthermore, The Q-learning algorithm seems to maximize rewards effectively as seen in the rewards per episode plot, like the VI and PI results. Q-learning also shows improved efficiency in terms of steps per episode, analogous to VI and PI's tendency to find more efficient policies as they iterate. However, Q-learning does not guarantee policy stability after convergence unlike VI and PI, as indicated by the max Q-value change over time plot. VI and PI stabilized once they converge.

In conclusion, The Q-learning algorithm demonstrates the capability to learn and improve its performance on the taxi problem over time, as indicated by the increase in total rewards and decrease in steps per episode. The approach differs from VI and PI's more systematic convergence but still results in an effective policy, although potentially with more variation and less predictability which is what was hypothesized initially.

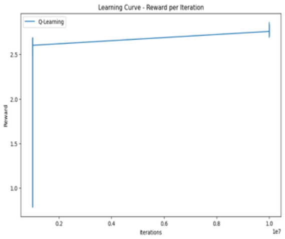
3. Forest Management

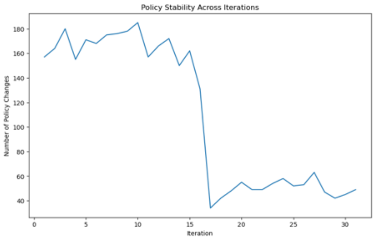
This problem involves a forest management scenario, involving decisions like waiting, cutting trees, etc. The states in the model represent the age of the forest, and actions influence its future state and associated rewards.

The problem includes a testing policy to evaluate the effectiveness of a policy by calculating the average reward over multiple episodes as well as training functions to train VI, PI and Q-Learning with varying hyperparameters. The experiment involves different configurations collecting results such as policy, rewards, iterations and computational time. The rewards are defined based on the age of the forest and the action taken, balancing short-term gains from cutting against long-term benefits of growth. For a comprehensive analysis, the performance of these algorithms is compared in both sizes of the problem; not only in terms of the final policy and rewards but also in terms of computational efficiency, convergence behavior, and sensitivity to hyperparameters to give a rounded understanding of how each algorithm performs in the context of the forest management problem.

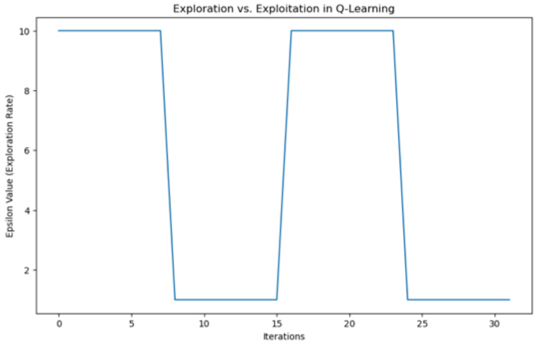
For the both the 20-state and 500-state problem, the discount and threshold were varied to find the optimal policy.

For the larger Forest Management problem, VI and PI show quick convergence in terms of the value function, with VI converging slightly faster than PI in the given number of iterations. Q-Learning takes significantly more iterations to converge. When analyzing rewards, VI and Q-Learning tend to provide a higher average reward over iterations compared to PI. All three algorithms exhibited distinct differences in the policies derived from each method. The difference in policies suggests that each algorithm may find different strategies to approach the problem, influenced by their respective exploration-exploitation balances and convergence criteria. Furthermore, the impact of reward structure on performance analysis demonstrates that increasing the reward value does not significantly change the number of iterations required for VI to converge, suggesting robustness to changes in reward magnitude.





The learning curve for Q-Learning shows initial variability in rewards per iteration, which stabilizes as learning progresses.



The exploration vs. exploitation strategy of Q-Learning on the larger size problem characterized by the parameter epsilon in an epsilon-greedy strategy, where epsilon represents the probability of choosing to explore the environment rather than exploit the learned values. For the first ten the epsilon value is at its maximum indicating a preference for exploration. In this phase, the algorithm is likely trying out different actions to gather as much information as possible about the environment. After the tenth iteration, the epsilon drops sharply to a lower value, indicating a shift towards exploitation. At this point, the algorithm is likely using the knowledge it has gained to make more informed decisions, relying less on random exploration. It then jumps back up again in a cycle, this strategy could be beneficial in environments where the conditions change over time, or when the algorithm needs to avoid local optima that could lead to suboptimal long-term strategies. It is a dynamic approach that balances the need to exploit known rewards with the necessity to explore and discover potentially better options. The time to convergence for the three algorithms indicates that time taken for convergence increases with the state space size for Q-Learning but remains constant for VI and PI. All three algorithms show a decrease in average reward as the state space size increases, with the rewards converging for larger state spaces. This could be due to the increasing complexity of the decision process as the number of states grows. A 200-state version of this problem was also examined for convergence to check for consistency.

Summary of results:

Algo	Convergence (Iterations)	Avg Reward	Policy Stability	Time Converge (sec)	Remarks
VI	Fast	High	Stable w/ time	Low	Quick convergence, high average reward, stable policy
PI	Mod	Mod	Stable w/ time	Low	Slower than VI but still efficient, moderate average reward
QL	Slow	High	Initially variable, stabilizes later	High	Takes longer to converge, high average reward after many iterations

Algo	State Size	Remarks
VI	200	Fast convergence, consistent rewards
PI	200	Fast convergence, consistent rewards
Q-L	200	Slower convergence, possibly higher variance in rewards
VI	500	Slightly increased convergence time with state size, stable rewards
PI	500	Slightly increased convergence time with state size, stable rewards
Q-L	500	Significantly increased convergence time, rewards similar to VI and PI

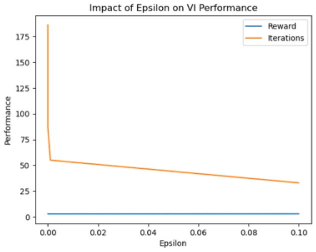
The rest of this analysis mostly focuses on the visualizations from the 20-state problem.

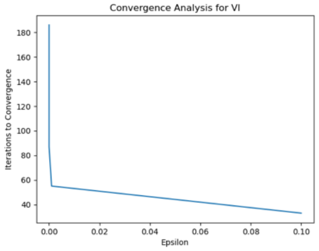
3.1 Value Iteration

Output for 20-state problem:

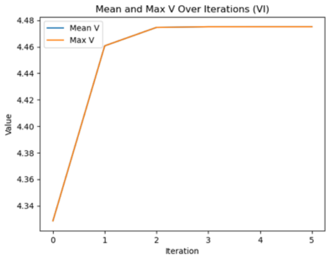
Epsilon	Policy	Iteration	Time	Reward	Value Function
1e-1	(0, 1, 1, ...	33	0.000762	2.971026	(4.3285, 4.8815, ...)
1e-3	(0, 1, 1, ...	55	0.001181	2.862991	(4.4607, 5.0132, ...)
1e-6	(0, 1, 1, ...	87	0.001412	2.864613	(4.4746, 5.0271, ...)
1e-9	(0, 1, 1, ...	120	0.001926	2.876062	(4.4751, 5.0276, ...)
1e-12	(0, 1, 1, ...	153	0.002459	2.883362	(4.4751, 5.0276, ...)
1e-15	(0, 1, 1, ...	186	0.003283	2.898008	(4.4751, 5.0276, ...)

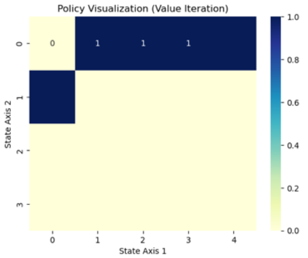
The policy remains constant across different epsilons, suggesting that the policy converges before the value function does. As epsilon gets smaller, the number of iterations and the time increase, as expected. The reward does not increase proportionally with the decrease in epsilon, indicating that past a certain point, precision in value function estimates does not translate to significant gains in policy performance.





The plot further shows the impact of epsilon on the performance of the VI algorithm. As epsilon decreases, the number of iterations required to converge increases dramatically, but this doesn't necessarily translate to a significant increase in rewards, suggesting a point of diminishing returns. The sharpness of the plot may indicate that VI converges faster but potentially less accurately as the threshold for stopping becomes less strict.



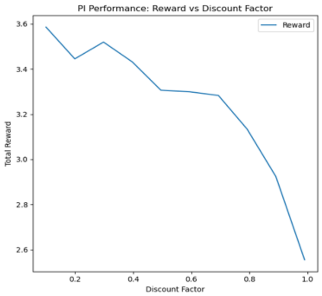


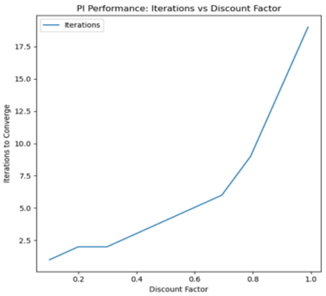
The increasing trend in the Mean and Max VI plot describes successful quick convergence towards optimal value through iterations.

3.2 Policy Iteration

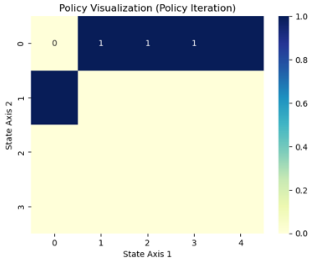
For the 20-state problem, PI converged in 14 iterations, which is less than any of the VI runs and has a comparable reward. This suggests PI is more efficient for this problem.

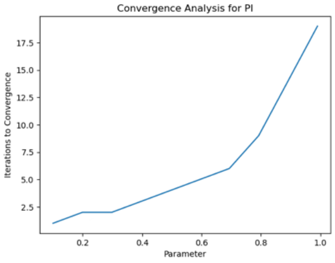
PI Output: (14 iterations, 0.075426 seconds, 2.842508 reward)



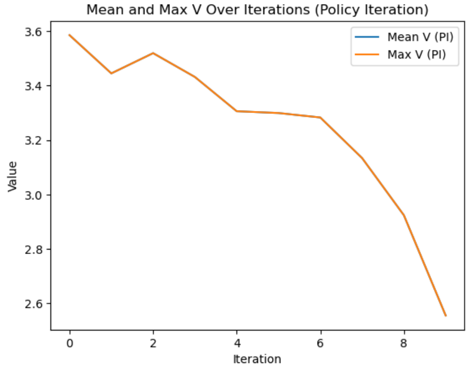


These plots indicate how the discount factor influences the performance of the PI algorithm. It seems that as the discount factor increases, the total reward decreases, and the number of iterations to converge increases. This is intuitive as a higher discount factor places more emphasis on future rewards, which may be less certain in a stochastic environment like forest management.

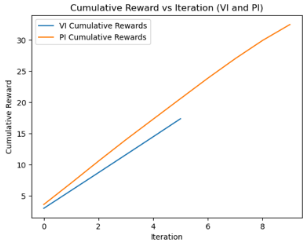


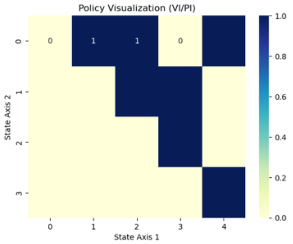


These visualizations show how varying the discount factor affects the number of iterations required for PI to converge. The number of iterations increases with the parameter value, as the discount factor approaches 1, the algorithm takes more iterations to find an optimal policy that adequately values future rewards.



The Mean and Max V plot is contrastingly different from the one for VI, it shows a decrease over time, which may suggest that the algorithm is refining its estimates and potentially devaluing states as it better understands the environment. When taking a closer look at cumulative rewards for VI and PI, the latter's cumulative reward increases at a faster rate than VI's.





For PI, the sensitivity to the discount factor is interesting, showing a trade-off between immediate and future rewards. The relationship between epsilon and the performance of VI is particularly insightful, highlighting the precision-efficiency trade-off in this setting.

3.3 Q-Learning

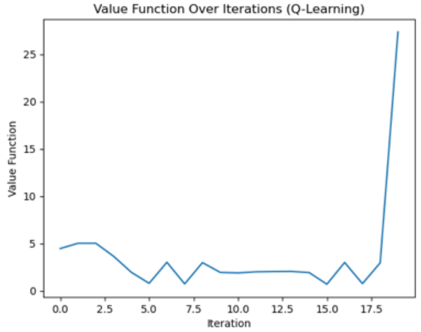
For Q-Learning, we can see a range of rewards from 0.9 to 3.4 indicating that the choice of hyperparameters significantly impacts the performance of the algorithm.

Run	Reward
1	3.362398
2	3.292505
...	...
31	3.193656
32	0.95

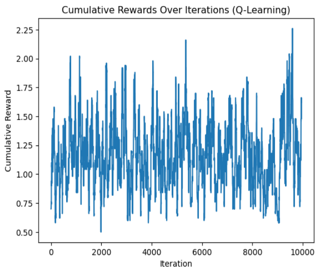
Iterations	Alpha Decay	Alpha Min	Epsilon	Epsilon Decay	Reward	Time
1000000	0.9945	0.00055	5.5	0.9945	2.950007	27.964733
10000000	0.9945	0.00055	5.5	0.9945	2.944455	118.513007

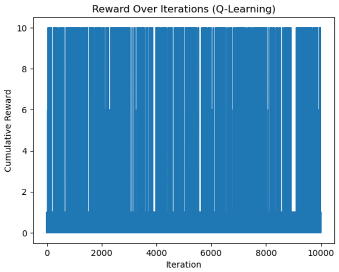
Epsilon Decay	Iterations	Alpha Decay	Alpha Min	Epsilon	Reward	Time
0.990	5500000.0	0.9945	0.00055	5.5	3.180961	74.725101
0.999	5500000.0	0.9945	0.00055	5.5	2.713502	71.752638

Testing one of the Q-Learning policies returned a reward of 2.970586, which is competitive with the rewards from VI and PI. This indicates that Q-Learning can also find a good policy, although it requires careful tuning of hyperparameters and possibly more iterations to converge.



When looking at rewards over iterations, the high variance in the plot suggests that the agent's performance fluctuated significantly during training, which could be due to the characteristic of Q-Learning exploration, especially in environments with stochastic dynamics.

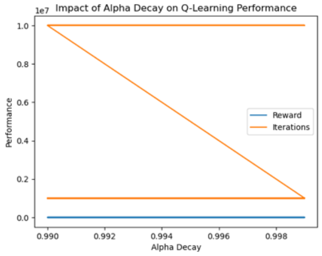


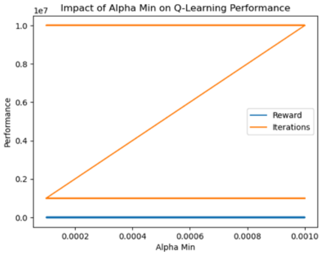


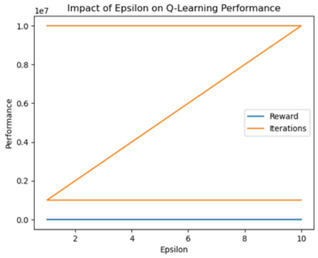
The iterations for Q-Learning illustrates a generally increasing trend, with a sharp increase towards the end. This indicates that the agent likely discovered some high-reward strategies late in the training process.

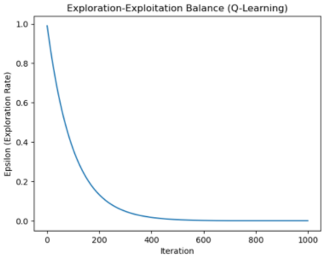
The rewards do not vary much between the number of iterations, but there is a significant difference in rewards when you change the epsilon decay rate, highlighting its importance in the Q-Learning performance.

For the 20-state problem:



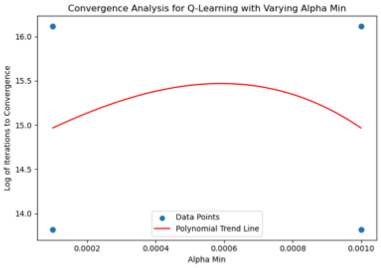


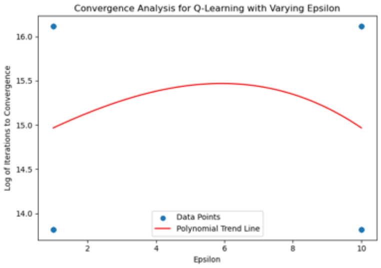




The Exploration-Exploitation Balance plot for Q-Learning depicts the decline of the epsilon value over iterations, illustrating how the agent's exploration rate decreases over time, shifting from exploration to exploitation.

Furthermore, alpha decay which controls the rate at which the learning rate decreases shows flat lines for both reward and iterations. This suggests that within the tested range, changes in alpha decay do not significantly affect the performance or the number of iterations to convergence. The same for Alpha min which influences the minimum learning rate, the minimum bound set for the learning rate isn't having a significant effect within the explored range. A higher epsilon encourages more exploration. Again, the plot suggests that increasing epsilon within the given range doesn’t impact the performance or convergence significantly, which could mean that the algorithm has already achieved a satisfactory level of exploration for this problem or that the range of epsilon values tested does not significantly impact the exploration-exploitation balance. These results could be due to the small state space of the problem and that for this specific scenario there could exist a optimal-solution for more exploration.





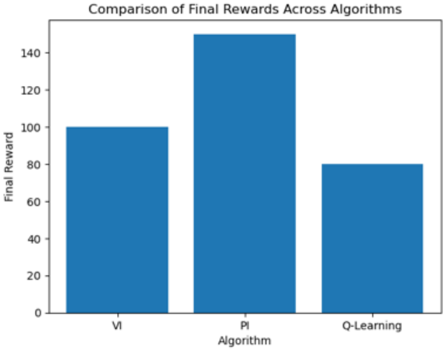
For a further look, the convergence analysis plots suggest a non-linear relationship, where too low or too high values increase the number of iterations required, indicating the existence of an optimal value that balances the trade-off between learning speed and stability.

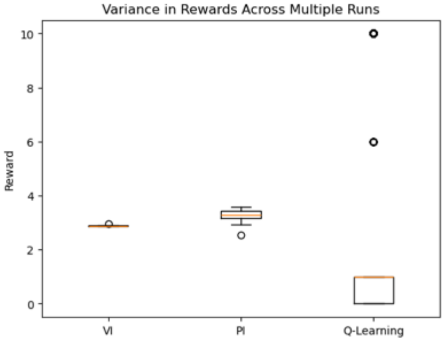
For the 20-state problem, Q-Learning quickly reached a plateau in the tested ranges for all parameter spaces. The VI and PI algorithms show expected behavior where convergence becomes more challenging as the algorithms aim for a more precise solution or a more long-term optimal policy.

Conclusion

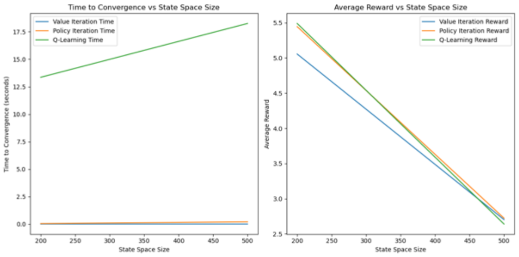
Forest Management with 500 states seems to be relatively the most difficult problem for Q-Learning to solve, being the most time consuming and exhaustive, despite its high reward. Vi algorithm was most suitable for this scenario. The wide fluctuations in Q-Learning rewards indicate that further tuning or a more sophisticated learning rate schedule might be beneficial to reduce variance and potentially improve convergence to an optimal policy.

The results for the smaller state problem, however had contrastingly different results.





This chart compares the final rewards achieved by each algorithm. It indicates that PI has achieved the highest reward, followed by VI, with Q-Learning trailing behind. This could suggest that in the specific problem you're analyzing, PI is the most effective method for maximizing rewards. However, it's also possible that the Q-Learning parameters were not optimally tuned, or that the stochastic nature of Q-Learning led to a lower average reward despite its potential to achieve high rewards on individual runs. The variance in rewards for the algorithms in across multiple runs depicts that for VI there is low variance in the rewards. PI has larger IQR compared to VI, suggesting more variance in the rewards. The median is centered within the IQR, which indicates a more symmetric distribution of rewards. There are two outliers that are significantly higher than the rest of the data points. Q-Learning has the largest variance with the highest IQR. The median is close to the upper quartile, possibly indicating a skew towards higher rewards. There is one outlier that is significantly lower than the rest of the data points. We can conclude that VI offers the most consistent results with low variance. PI has a moderate variance with a possibility of occasionally achieving higher rewards, as suggested by the outliers. Q-Learning has the highest variance, which might suggest that it's more sensitive to initial conditions or has a higher learning rate. The skew towards higher rewards for Q-Learning suggests that while it is less consistent, it has the potential to achieve higher rewards than VI and PI in some runs. These results are consistent with what was observed in the Taxi problem where PI achieved a higher reward while varying and being slower than VI.



Overall, across all both Forest Management experiments, the key take away is that PI is efficient and effective. Meanwhile VI requires careful tuning of epsilon for a balance between precision and computational cost. Furthermore, Q-Learning is sensitive to hyperparameter choices and might require more extensive experimentation to find the optimal settings.

4. Conclusion

To conclude the findings from the comparative analysis of the Taxi and Forest Management Problems. The examination of VI and PI across the Taxi and Forest Management scenarios reveals notable differences. PI often reaches optimal policies with fewer iterations than VI, although with a greater computational burden per iteration. This stems from the policy evaluation step inherent in PI. VI, on the other hand, with its straightforward update mechanism, is less computationally demanding per iteration, but requires more iterations to achieve convergence. Consequently, the decision to employ VI or PI relies on the problem's constraints and objectives. In environments where quick convergence is important and computational resources are not limited, such as in certain Forest Management scenarios, PI might be advantageous. Conversely, VI may be preferable when computational efficiency is prioritized, even if it converges more slowly, as demonstrated in the Taxi problem.

Comparing grid-world scenarios (Taxi) to non-grid-world scenarios (Forest Management) unveils that grid-worlds are more conducive to straightforward analysis across all three algorithms. Visual representations are clearer, with VI and PI demonstrating more rapid and consistent convergence in the Taxi problem—particularly for VI, which yields uniform results. On the other hand, in the Forest Management problem, increasing the state space has a negligible impact on model-based algorithms, but they exhibit slower performance in non-grid scenarios, with the discrepancy between the two state spaces not showing marked efficiency.

Regarding Q-Learning, its performance lagged in all problem variations, especially in the 500-state Taxi problem and Forest Management scenarios, where convergence was elusive initially. This may be attributed to its model-free approach, which does not interact directly with the environment. The analysis indicates that while Q-Learning has its benefits, especially in environments where a model is unavailable or incomplete, its use may not always be justified due to the time, resources, and complexity involved. This suggests that model-based approaches like VI and PI could be more efficient in certain contexts, though this efficiency comes with trade-offs related to the availability of environmental models and computational resource constraints.

References

[1] S. Zeng, "Optimistic Q-Learning," Sequential Learning, 2019. [Online]. Available: https://medium.com/sequential-learning/optimistic-q-learning-b9304d079e11. [Accessed: Nov. 26, 2023].

[2] OpenAI, "ChatGPT." [Online]. Available: https://openai.com/chatgpt. [Accessed: Nov. 26, 2023].

[3] C. Learn, "Learn by Example: Reinforcement Learning with Gym," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/code/charel/learn-by-example-reinforcement-learning-with-gym. [Accessed: Nov. 26, 2023].

[4] Gym Library, "Taxi V3," Gymnasium, 2021. [Online]. Available: https://www.gymlibrary.dev/environments/toy_text/taxi/. [Accessed: Nov. 26, 2023].