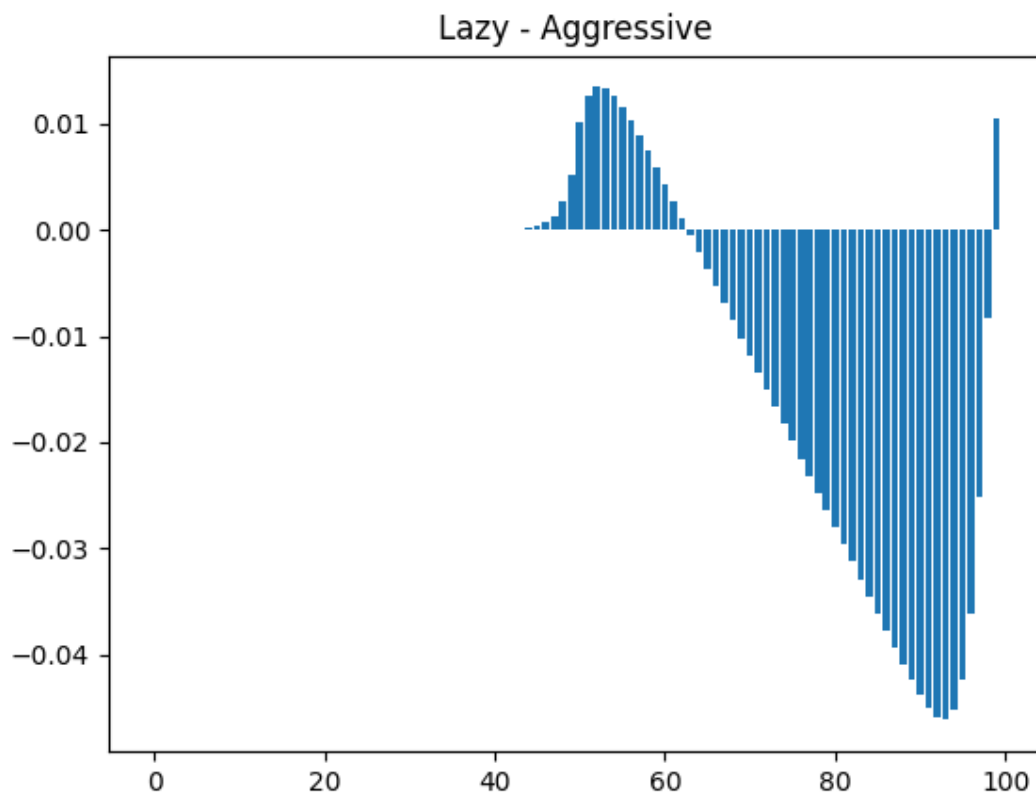


Reinforcement Learning Problem Set 1

Problem 1

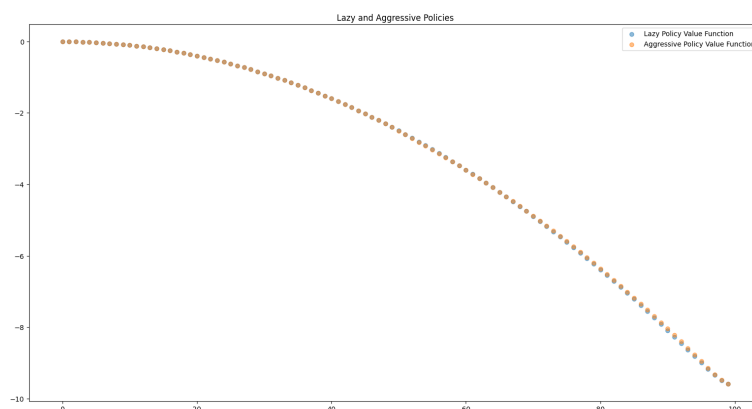
Below we see the difference between the value function of Lazy Policy and Aggressive Policy.



Difference at timestep 50 is 0.010109890229023755 : Lazy Policy is better

Difference at timestep 80 is -0.027995644663423747 : Aggressive Policy is better.

The plot below shows both value functions plotted together.



All the code used for the project can be found here: <https://github.com/aakash94/UPFRL/tree/main/src/hw1>

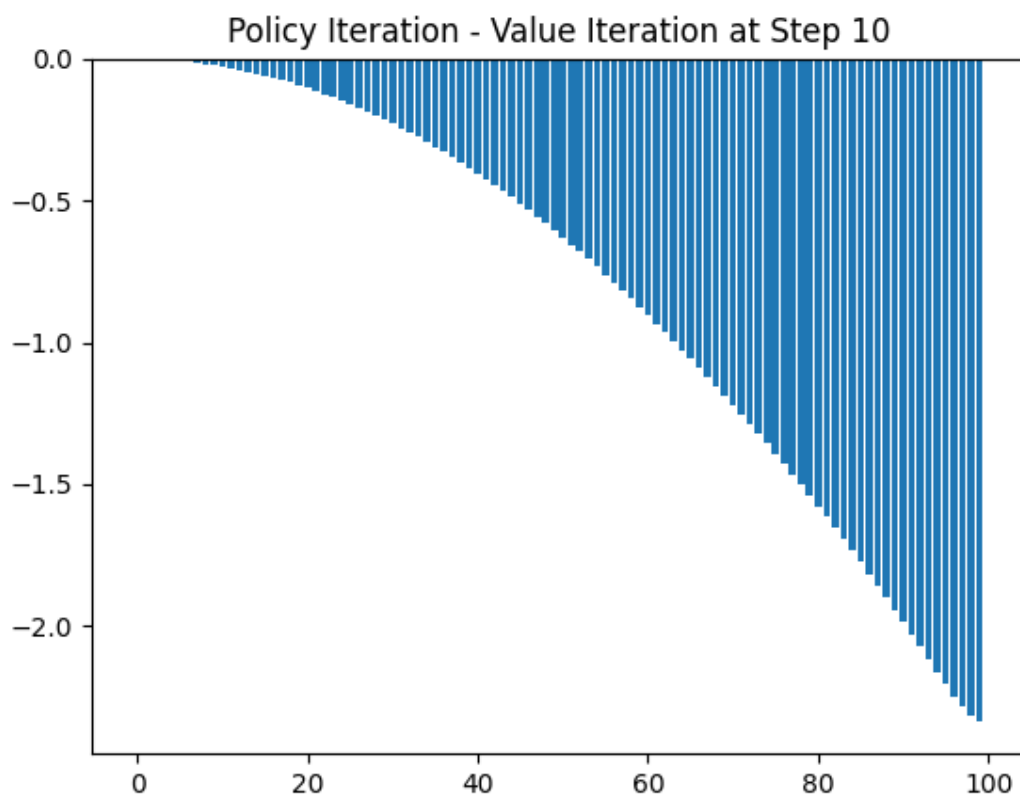
Problem 2

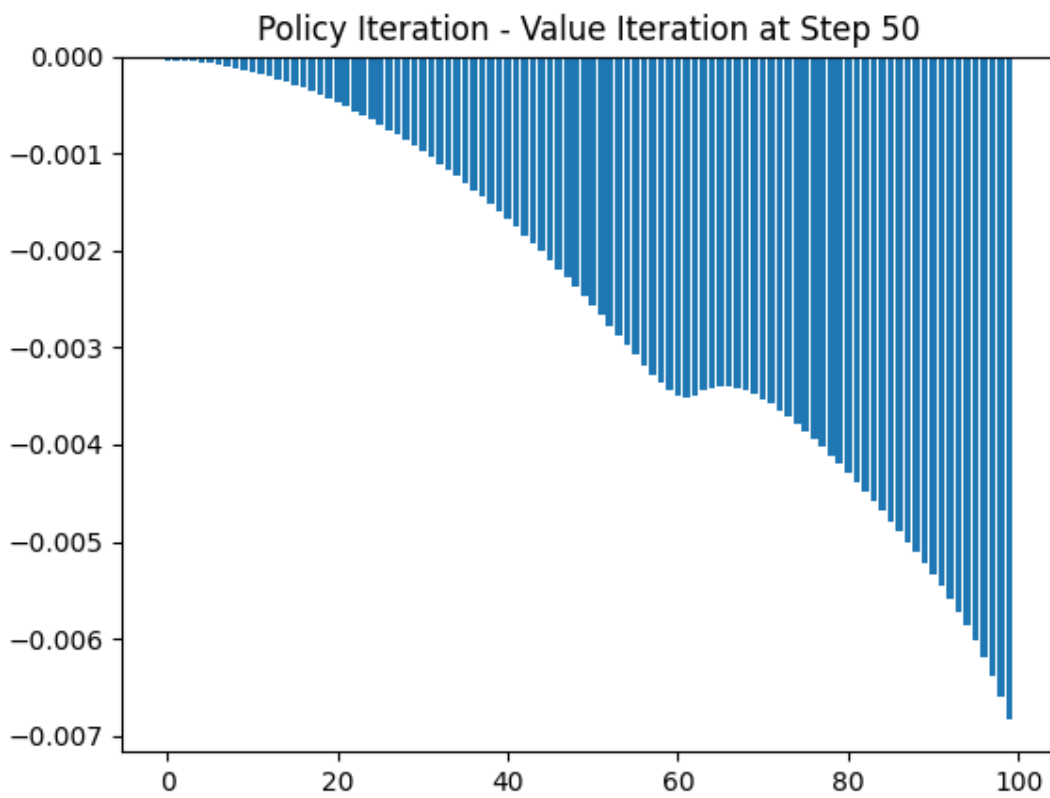
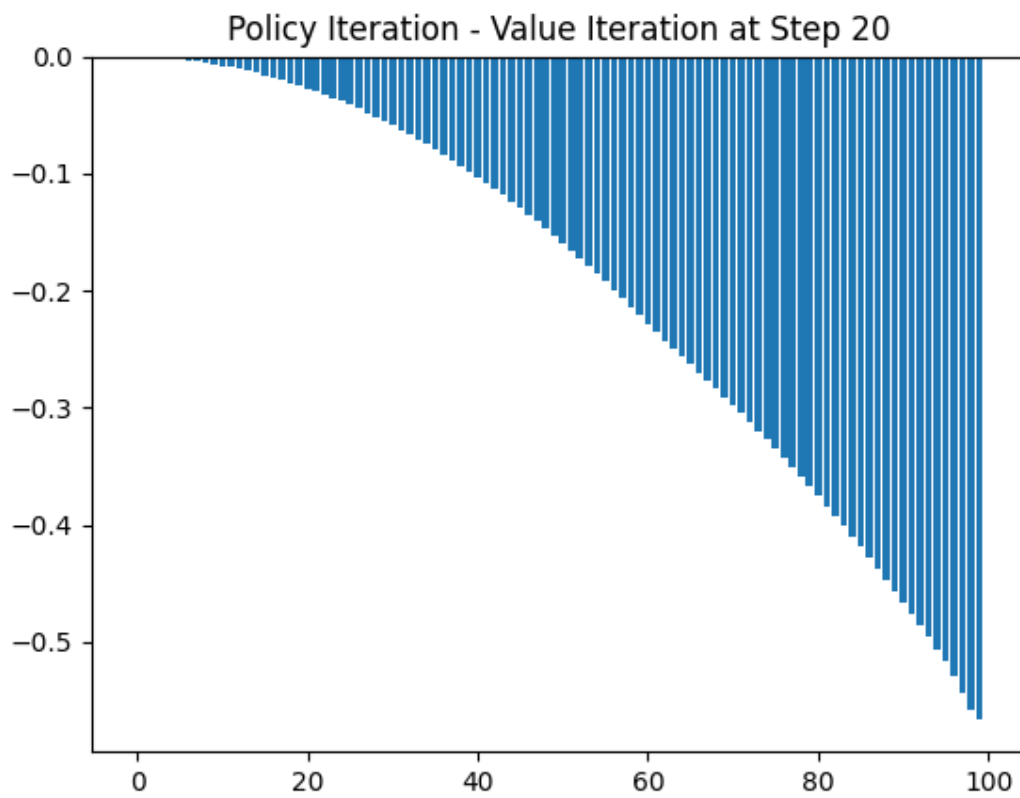
Policy iteration took 1 iterations and 0.96875 s

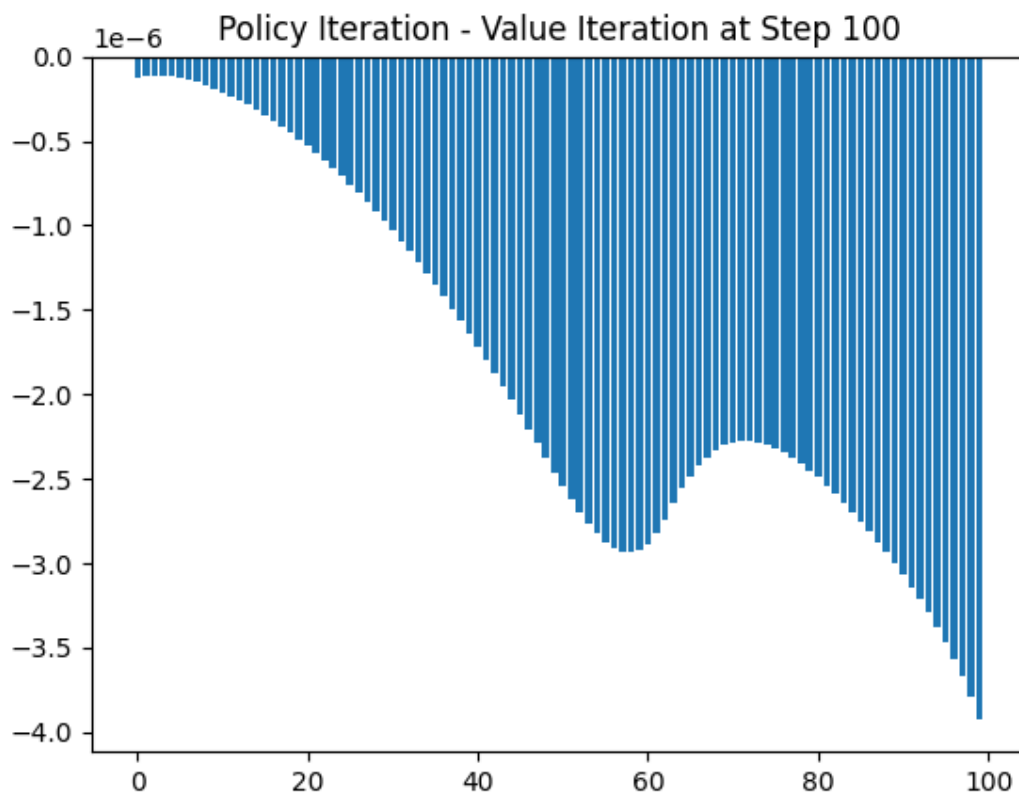
Value iteration took 127 iterations and 0.140625 s

Value Iteration time for 100 steps is 0.109375s

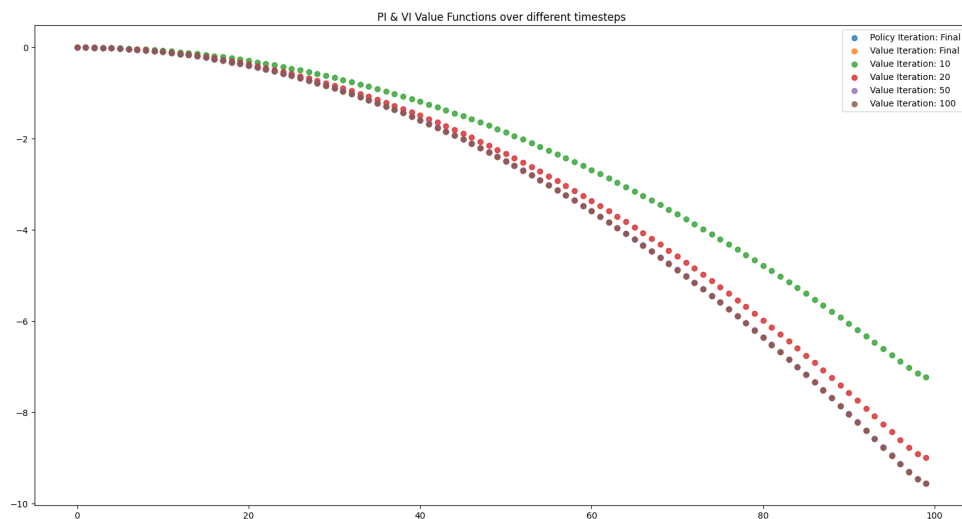
Below are plots showing the difference between the value function of Policy Iteration and Value Iteration at different timesteps.





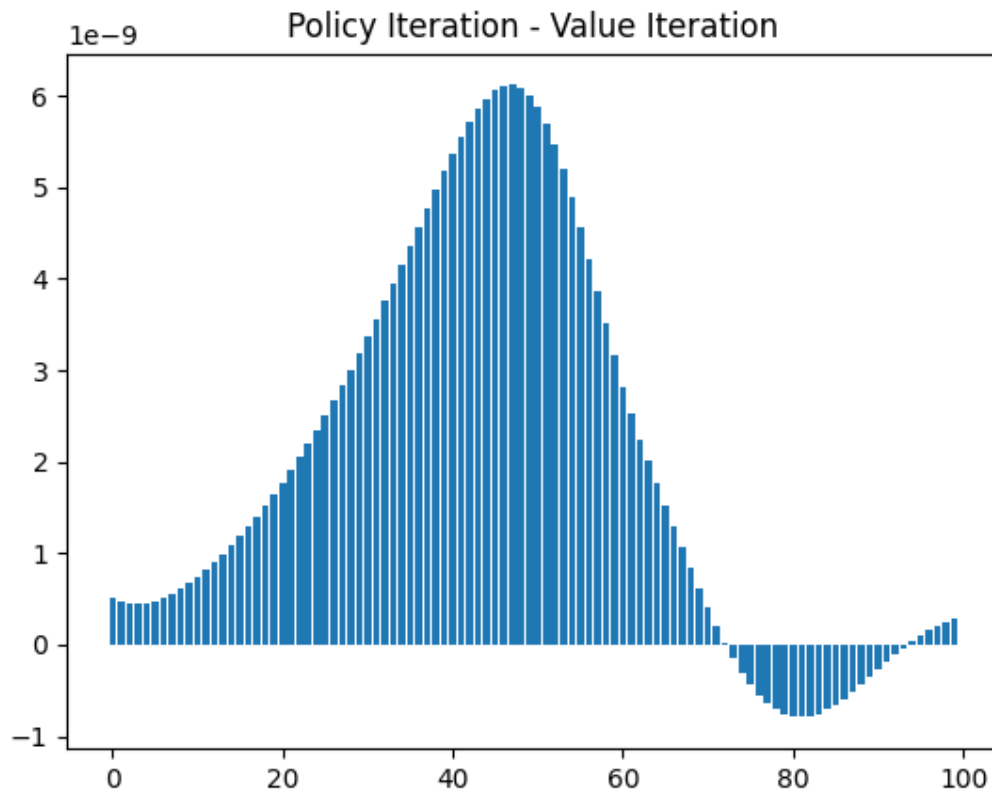


Here is another plot showing all of the values together.



All the code used for the project can be found here: <https://github.com/aakash94/UPFRL/tree/main/src/hw1>

We see that the difference between the Policy Iteration and Value Iteration's Value functions get smaller with timesteps and finally converge to the optimal value function. The difference between the final Policy Iteration Value function and final Value Iteration Value function is shown below.



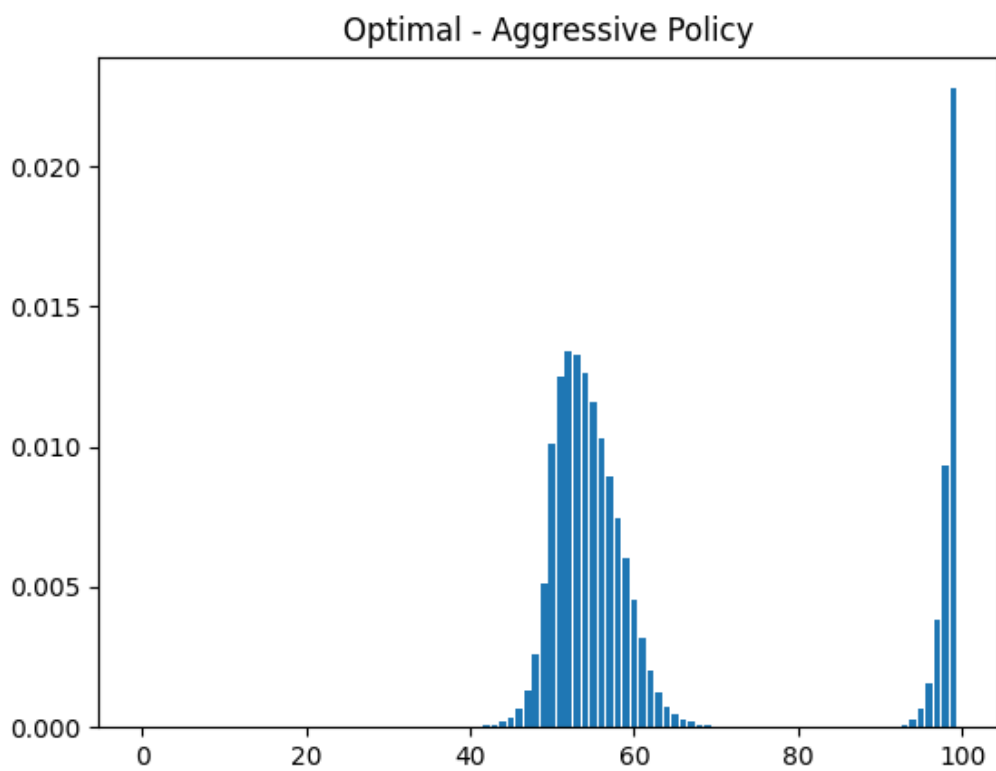
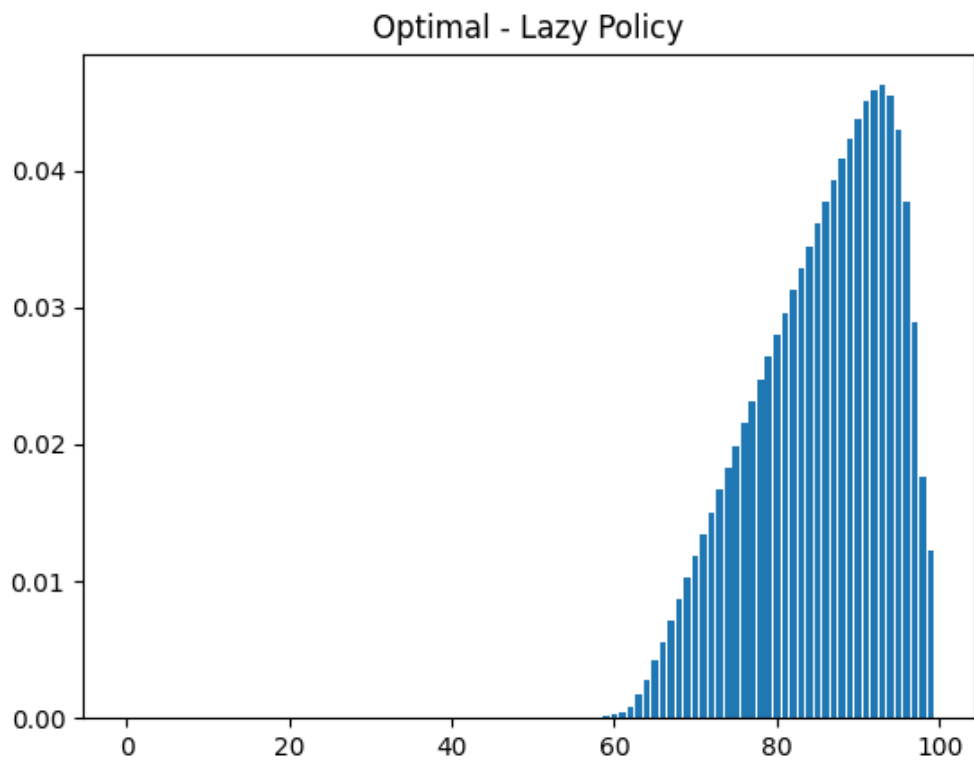
Policy Iteration and Value Iteration both converge to the same value function (within margin of error that is $\theta=1e-8$).

Value iteration takes more steps, but the steps are much faster, hence taking less time in total. Whereas Policy Iteration takes only one longer step to converge.

The policy evaluation step done for Policy Iteration is an expensive step that causes Policy Iteration to take longer. Since Value Iteration does not have a policy evaluation step, it can execute an iteration much faster.

However it is worth noting that in the case of value iteration, we wait for the value function to converge before taking a policy out of it. In the case of policy iteration we create intermediate policies in every step, and stop as soon as we find that the newer policy does not improve on the older one.

Comparison with Problem 1



Conclusion

Looking at the difference between the value function of Optimal Policy and the Lazy and Aggressive Policies, we can conclude that the Optimal Policy's Value function is either greater than or equal to the value function of the other policies at every state. Hence, we can state that the Optimal Policy strictly improves over other policies.

The Plots below show the action taken by different policies in different states.

