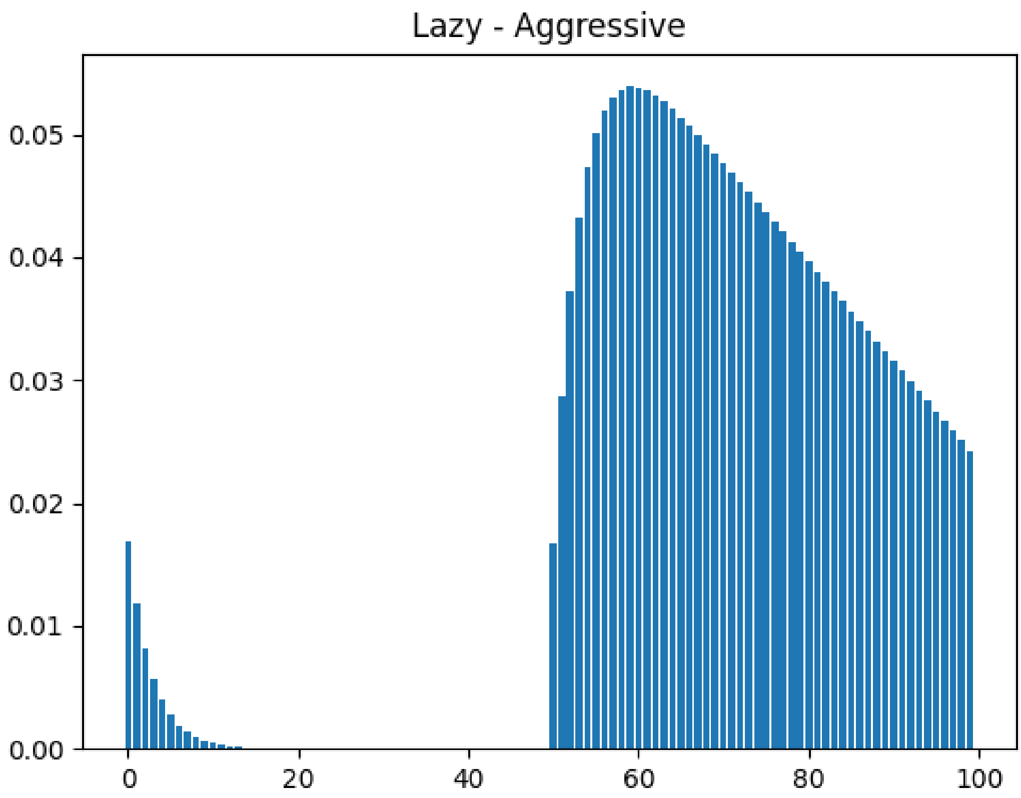
Reinforcement Learning Problem Set 1

Problem 1

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



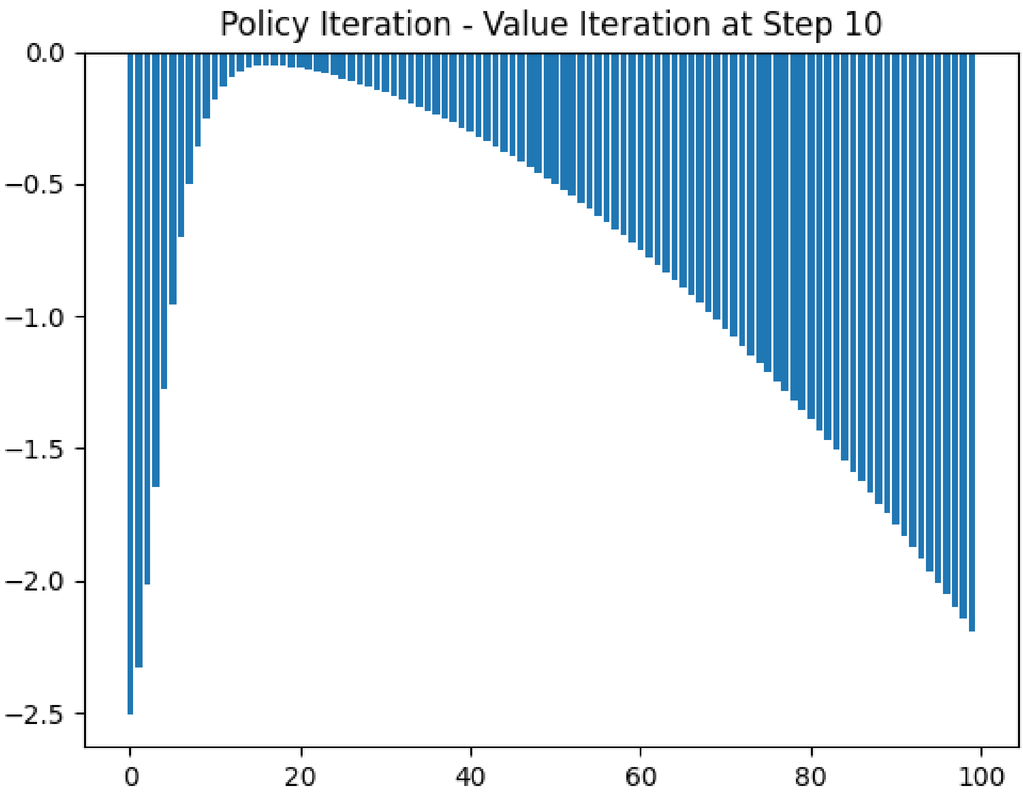
Difference at timestep 49 is -1.4633565470489884e-09 ≈ 0
Difference at timestep 50 is 0.016692962355814522
Difference at timestep 80 is 0.03964724581749124

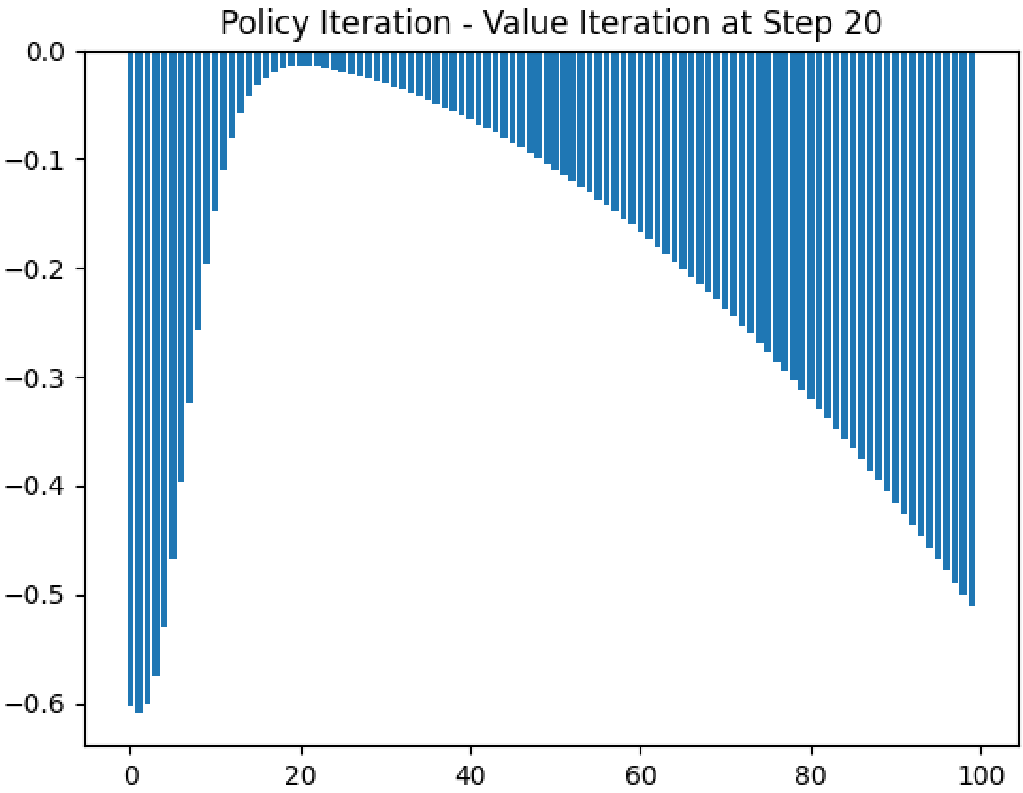
We can see that the Lazy Policy is strictly better than the Aggressive Policy across all states.

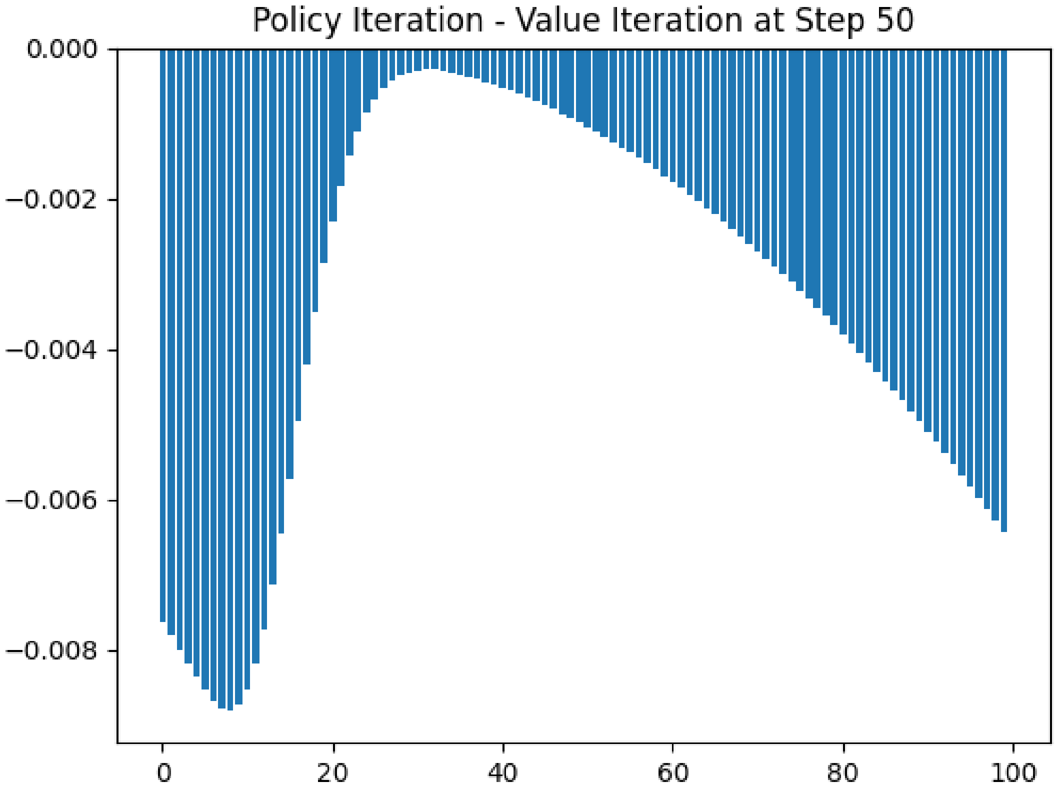
Problem 2

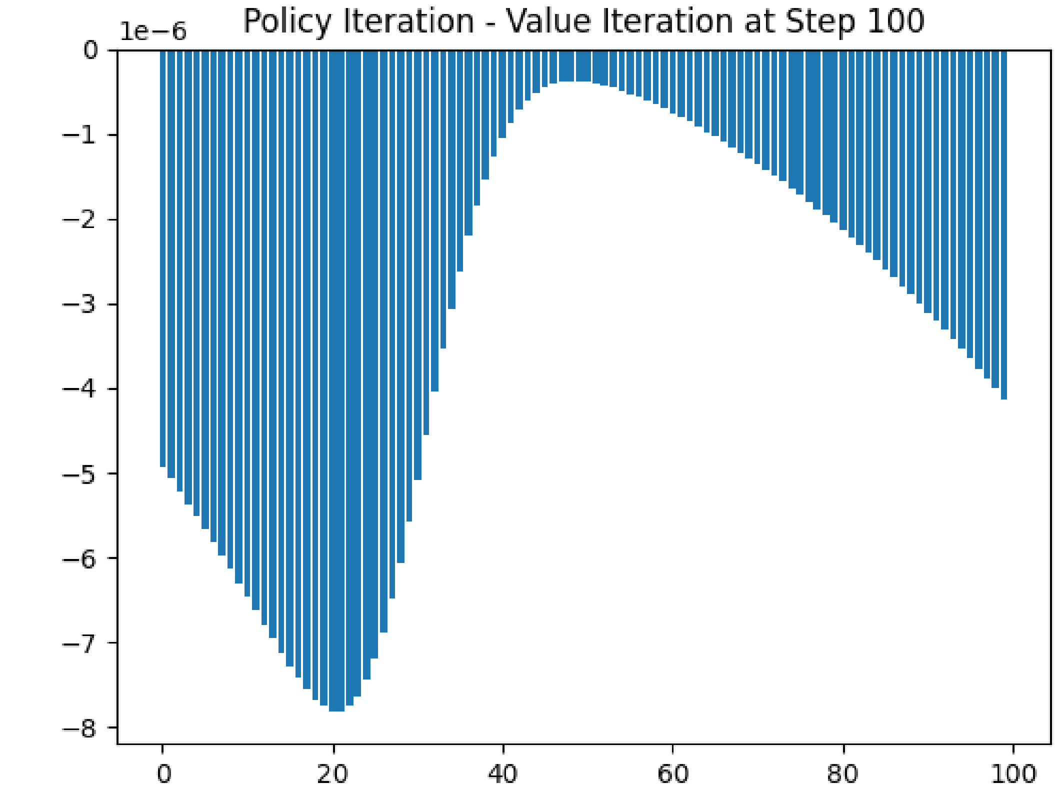
Policy iteration took 1 iterations and 0.84375 s
Value iteration took 134 iterations and 0.125 s
Value Iteration time for 100 iterations is 0.09375 s

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

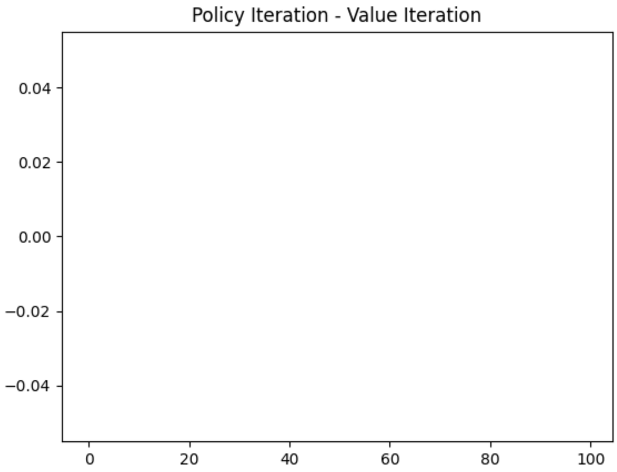

Policy Iteration - Value Iteration at Step 10
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100


Policy Iteration - Value Iteration at Step 20
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
0
20
40
60
80
100


Policy Iteration - Value Iteration at Step 50
0.000
−0.002
−0.004
−0.006
−0.008
0
20
40
60
80
100

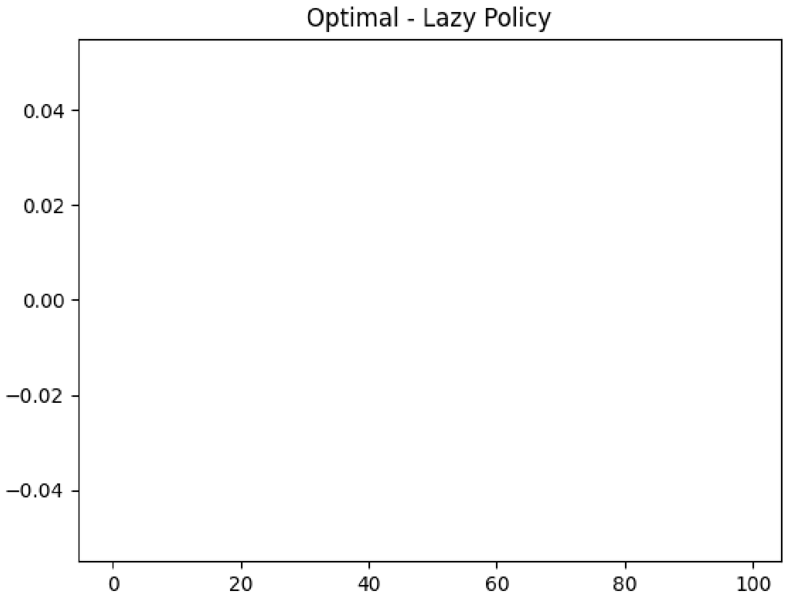

Policy Iteration - Value Iteration at Step 100
1e−6
0
−1
−2
−3
−4
−5
−6
−7
−8
0
20
40
60
80
100

We see that the difference between the Policy Iteration and Value Iteration 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.

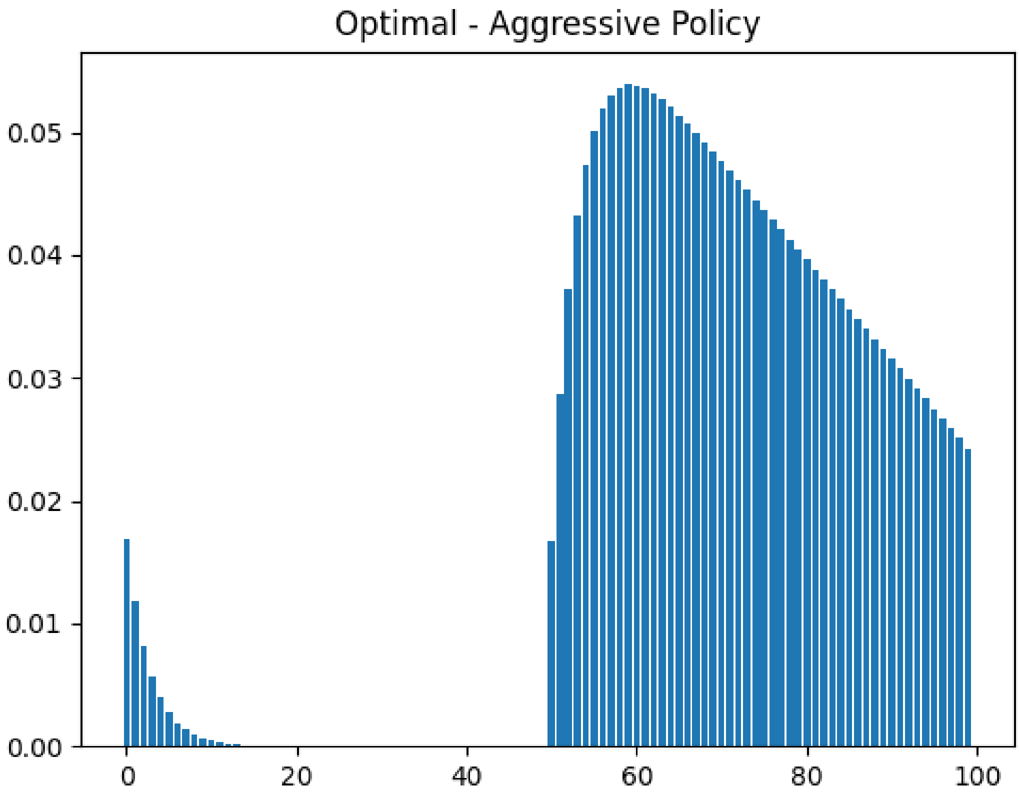


Policy Iteration and Value Iteration both converge to the same value function.
Value iteration takes more steps, but the steps are much faster, hence taking less time in total. Whereas Policy Iteration takes one longer step to converge.

Comparison with Problem 1



We can see here that the Lazy Policy is an Optimal Policy.



We can also see that Aggressive Policy is not optimal and that the Optimal Policy strictly improves over it.

Conclusion

Given the fixed parameters for the problem statement, the Optimal Policy is the Lazy Policy.