RL Lab 2 - Optimal Policies with Dynamic Programming

Adonis Jamal

1 Policy Evaluation

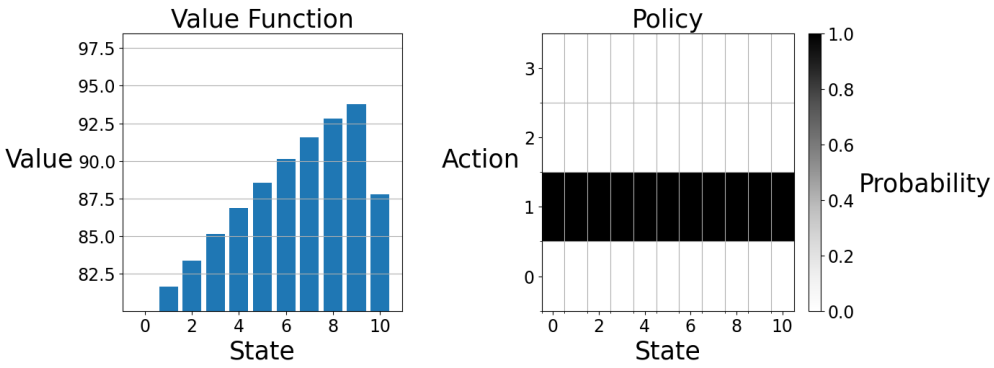

Figure 1: Policy Evaluation: Value function histogram and Action-State policy probabilities heatmap.

Figure 1 shows the results of policy evaluation, the city's baseline policy.

Observing the value function histogram, we see that the values increase as we move from state 0 (empty) to state 9, but we notice a drop at state 10 (full). The increase in values from state 0 to state 9 reflects the reward structure, where social welfare is higher when more parking is being used. The drop at state 10 however happens because of the specific penalty, where the city prefers having at least one spot open. A complety full parking lot is less valuable than one that is almost full.

The action-state policy heatmap shows a vertical line at action 1, while the probabilities for all other actions are zero. This indicates that we have a deterministic policy, where in every single state (from 0 to 10, empty to full), the agent (city) always chooses action 1 with probability 1, charging the same price regardless of the current state of the parking lot. This is a simple policy that does not adapt to the number of cars currently parked, which is why we see the same action being chosen across all states.

2 Policy Iteration

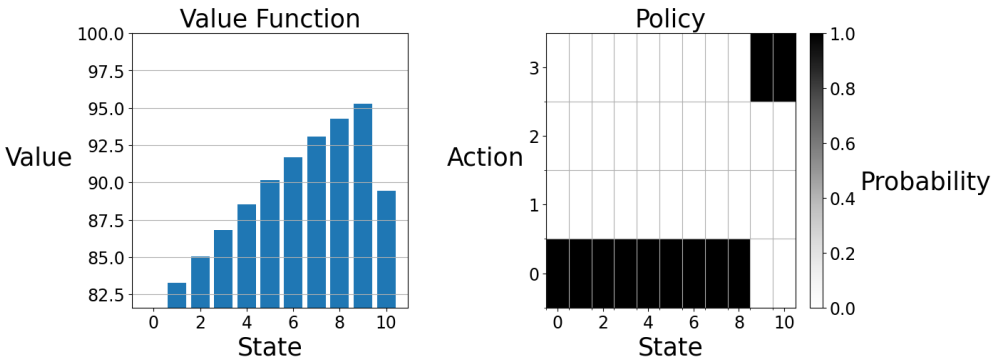

Figure 2: Policy Iteration: Value function histogram and Action-State policy probabilities heatmap.

Figure 2 shows the results of policy iteration, which finds the optimal strategy found by iteratively improving the policy.

The value function histogram shows a similar shape to the one from policy evaluation, with values increasing from state 0 to state 9 and a drop at state 10. However, the values are higher across all states compared to the policy evaluation results. This indicates that the new policy found by policy iteration is extracts more social welfare from the parking lot, as it is more effective at charging prices that encourage optimal usage of the parking spaces. We still see the drop at state 10, which is expected given the penalty for having a full parking lot.

The action-state policy heatmap for policy iteration now shows that action 0 (charging a lower price) is chosen with probability 1 for states 0 to 8, while action 3 (highest price) is chosen with probability 1 for states 9 and 10. Our optimal strategy now adjusts prices based on the current state of the parking lot. When the lot has space (states 0 to 8), the policy charges the lowest price to encourage more cars to park. This moves the system toward high-reward high-occupancy states. Howver when the lot is almost full (state 9), the policy charges the highest price to discourage additional cars from parking, which helps avoid the penalty associated with a full parking lot (state 10). This adaptive pricing strategy is more effective at maximizing social welfare compared to the baseline policy.

3 Value Iteration

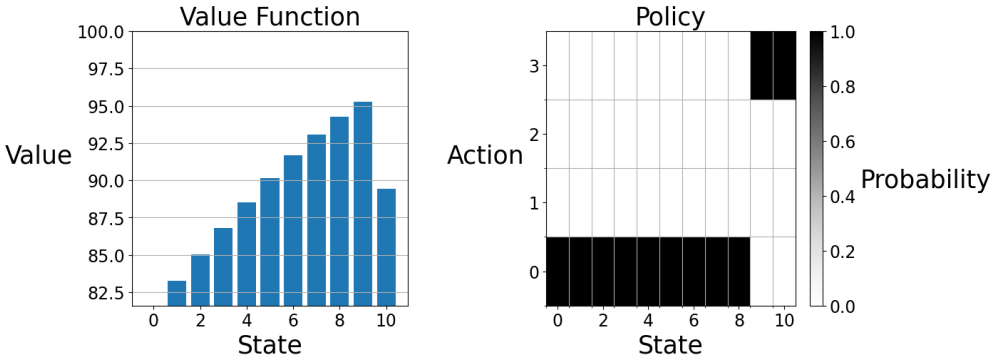

Figure 3: Value Iteration: Value function histogram and Action-State policy probabilities heatmap.

Figure 3 shows the results of value iteration, which finds the optimal strategy by iteratively improving the value function.

The value function histogram for value iteration is identical to the one from policy iteration, which is expected since both methods should converge to the same optimal value function. The values increase from state 0 to state 9 and drop at state 10, reflecting the same reward structure and penalty as before.

The action-state policy heatmap for value iteration is also identical to the one from policy iteration, showing that action 0 is chosen for states 0 to 8 and action 3 is chosen for states 9 and 10. This confirms that both methods have converged to the same optimal policy, which adjusts prices based on the current state of the parking lot to maximize social welfare.