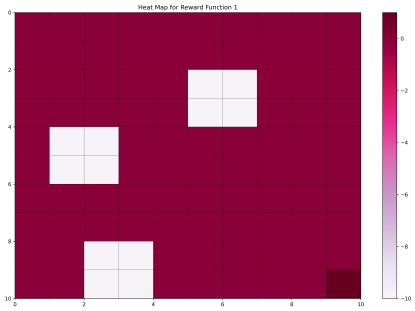
ECE232E Project3 Report

Luke Lim Angie Fan

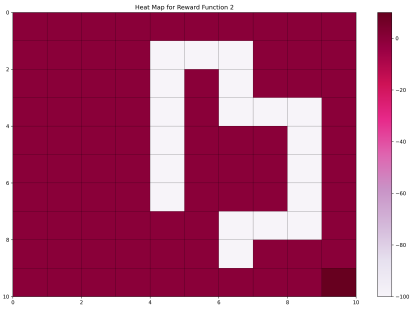
May 2024

Question 1

In this question, we utilize heatmaps to visualize both Reward Function 1 and Reward Function 2, including the color scale for each map.



((a)) Heat Map for Reward Function 1



((b)) Heat Map for Reward Function 2

Figure 1: Heat Maps for Ground-Truth Reward Functions

Question 2

This question outlines the necessary parameters: 100 states, 4 actions, $w = 0.1$, discount factor $\gamma = 0.8$, reward function from Question 1, and a threshold of 0.01.
Given these parameters, we need to perform the initialization and estimation for the value iteration procedure. The resulting table, showing the optimal value of each state, is presented below:

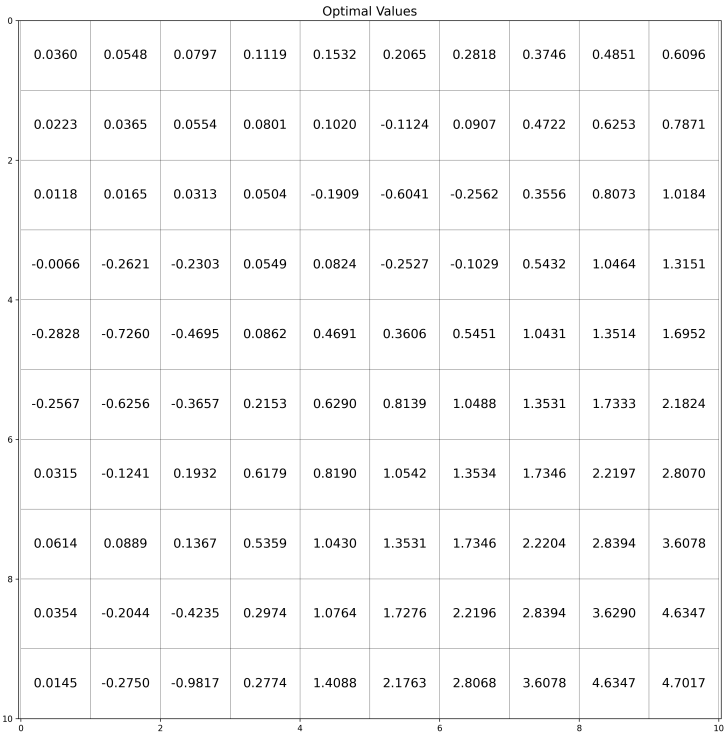


Figure 2: Optimal Values of the States for Reward Function 1

We observed that the algorithm converges in 22 steps. Figure 3 below depicts the snapshots of the state values for 5 different steps (step 4, step 8, step 12, step 16, step 20).

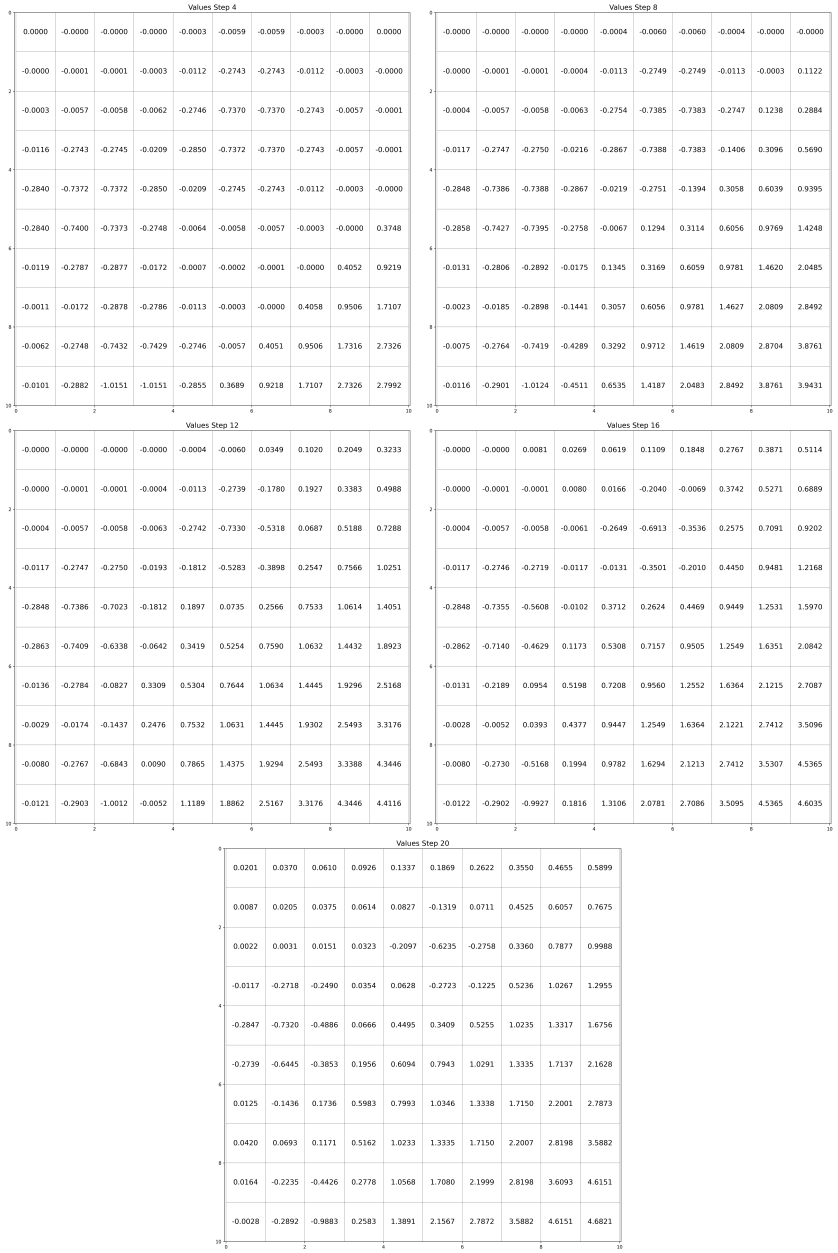


Figure 3: State Vales in Value Iteration for Different Steps (4,8,12,16,20)

Figures 2 and 3 show that state 99 (in the lower right corner) has the highest optimal value because it offers the highest reward for reward function 1. As you move away from state 99, the optimal value of other states decreases gradu-

ally. Additionally, states that yield negative rewards also have negative optimal values. The states near these negative-reward blocks have lower optimal values compared to those near state 99. This means that as an agent moves towards states with lower rewards, the optimal values of neighboring states decrease. Conversely, moving towards states with higher rewards increases the optimal values of neighboring states.

Furthermore, Figure 3 snapshots reveal that most states initially have a value of 0, creating a sparse $V(s)$ matrix. As steps progress, the values of states near state 99 (the highest reward state) gradually increase, while values of states near negative-reward states gradually decrease. Since state 99 has the highest reward, its value and those of neighboring states rise faster. As the algorithm gets closer to convergence, the sparsity in $V(s)$ reduces, with most states achieving a non-zero optimal value. States near state 99 get high positive values, while those near negative-reward blocks get negative values. The rate of change in state values is faster in the beginning, following a logarithmic rate of convergence until $\Delta < \epsilon$.

Question 3

To present Figure 2 in a more visual manner, we also generate its heatmap across the 2D grid. The heatmap is displayed below:

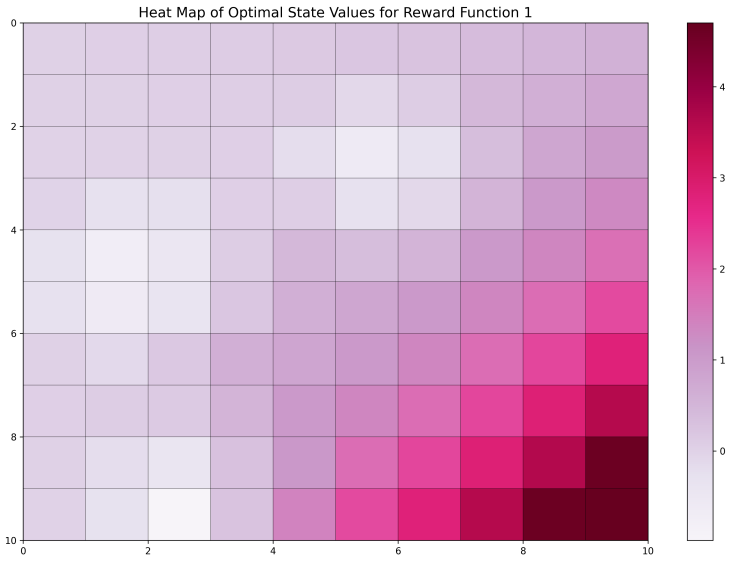


Figure 4: Heat Map for Optimal State Values (Reward Function 1)

Question 4

From the heatmap in Figure 4, we can infer that lighter regions indicate states with low optimal values, while more red regions correspond to states with high optimal values. State 99, having the highest reward, exhibits the highest optimal value. As one moves away from state 99, the optimal values decrease gradually. States near blocks with negative rewards also show lower optimal values compared to those near state 99. This means that as an agent moves towards states with low rewards, the optimal values of adjacent states decrease, and as it moves towards states with high rewards, the optimal values increase.

The heatmap reveals a gradual decay of optimal values around the state with the highest reward, due to the discount factor in the Bellman equation. The patterns in the heatmap for reward function 1 in Figure 1 match the heatmap of optimal values, showing three blocks of negative rewards. These blocks correspond to similar regions in the state space. The area near the highest reward state is also more red, suggesting that one can infer the reward function by observing how an agent or expert behaves, which is a principle used in inverse reinforcement learning (IRL).

Question 5

In this section, we perform the computation step of the value iteration algorithm to determine the optimal actions. We then visualize these actions using arrows in the state table. The resulting figure is displayed below:

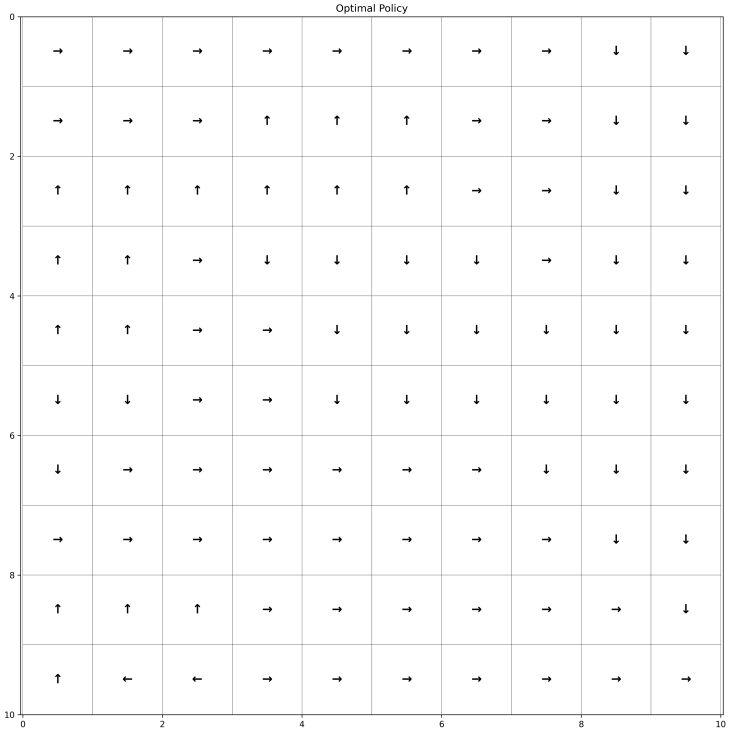


Figure 5: Optimal Action for Reward Function 1

Figure 5 reveals that all arrows point towards state 99, which has the highest reward of all states. The heatmap in Figure 4 supports this by showing that arrows at each state point towards the state that will eventually maximize the expected reward. The arrows guide the agent away from areas with negative rewards (lighter regions on the heatmap) towards more red regions, aligning with the goal of maximizing cumulative rewards within a finite horizon, thus targeting state 99 due to its highest reward.

Additionally, the agent can determine the optimal action at each state by observing the optimal values of its neighboring states. This is demonstrated by the arrows pointing towards the neighboring state with the highest optimal value, as seen in Figure 2. Even if the agent does not know future optimal values, it can still develop an optimal policy by observing local optimal values. This approach relies on the fact that the optimal policy depends on the optimal val-

ues of neighboring states, the transition matrix, and rewards. Thus, the agent selects actions that maximize expected rewards based on local observations.

Question 6

Starting from this question, we will use reward function 2 (from Question 1) as our reward function. All other parameters remain unchanged. Following the same initialization and estimation procedure as in Question 2, a table has been created to demonstrate the optimal value of each state. The table is shown below:

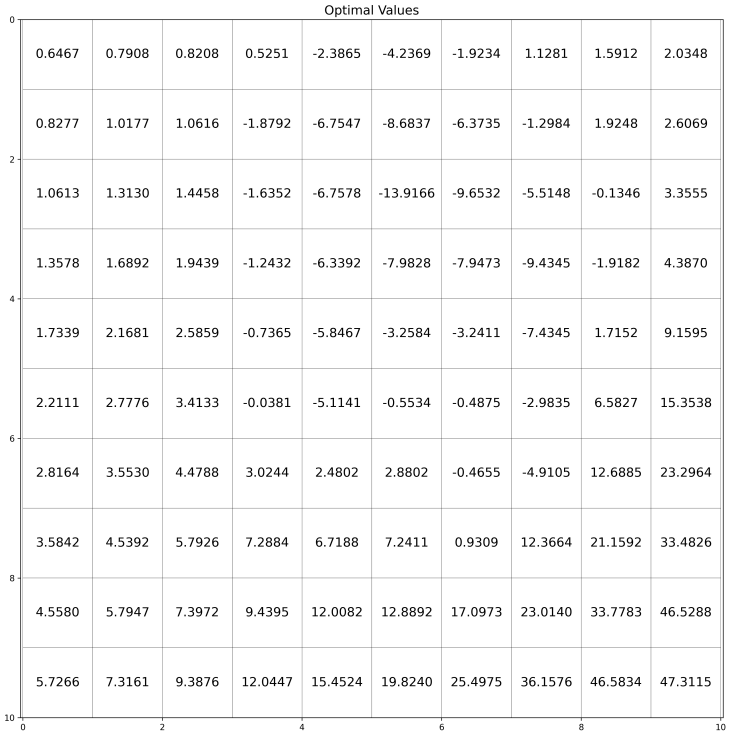


Figure 6: Optimal Values of the States for Reward Function 2

Figure 6 illustrates that state 99 (located in the lower right corner) has the highest optimal value, as it provides the highest reward under reward function 2. Moving away from state 99, the optimal value of each state gradually diminishes. States that yield negative rewards display negative optimal values, and their neighboring states also have lower optimal values compared to those near the state with the highest reward. Essentially, as an agent approaches states with low rewards, the optimal values of neighboring states decrease. Conversely, as an agent approaches states with higher rewards, the optimal values of the neighboring states increase.

Question 7

To present Figure 6 in a more visual manner, we have also generated its heatmap across the 2D grid. The heatmap is displayed below:

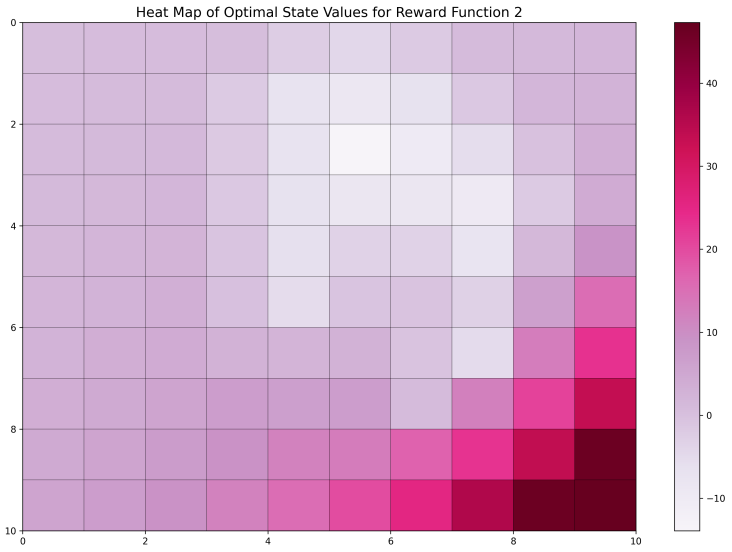


Figure 7: Heat Map for Optimal State Values (Reward Function 2)

From the heatmap in Figure 7, we can observe that lighter regions represent states with low optimal values, while more red regions correspond to states with high optimal values. State 99 stands out with the highest reward and optimal value. As we move away from state 99, the optimal value of each state gradually decreases. States near areas with negative rewards also have lower optimal values compared to those near state 99. This means that as an agent moves towards states with lower rewards, the optimal values of neighboring states decrease, while moving towards states with higher rewards increases the optimal values. The further a state is from the highest reward state, the lower its value.

The heatmap of reward function 2 in Figure 1 shows patterns that are visible in Figure 7. Negative rewards form a snake-like chain in the state space, which is also evident in the heatmap where the lightest regions represent states with the lowest rewards. Additionally, the area near the state with the highest reward is more red, suggesting that one can infer the reward function by observing how the environment or an expert behaves, a principle used in inverse reinforcement learning (IRL).

State 52, shown as the lightest in the heatmap, has the lowest optimal value. This is because state 52 is surrounded by states with negative rewards on three

sides, with only one path that does not penalize the agent. As a result, value iteration encourages the agent to avoid this state, as it has the highest probability of landing in a state with negative rewards among all states.

Furthermore, some states with negative rewards are closer to state 99 than others, yet the farther states have higher optimal values than those near the negative-reward chain. This indicates that proximity to a high reward state does not necessarily guarantee a high optimal value, especially if most neighboring states have negative rewards. This pattern is particularly evident for states in the upper left corner, which have higher optimal values than those near the negative-reward chain. The gradual decay of optimal values around the highest reward state is due to the discount factor in the Bellman equation, which reduces the impact of future rewards.

Question 8

In this section, again we perform the computation step of the value iteration algorithm to determine the optimal actions. We then visualize these actions using arrows in the state table. The resulting figure is displayed below:

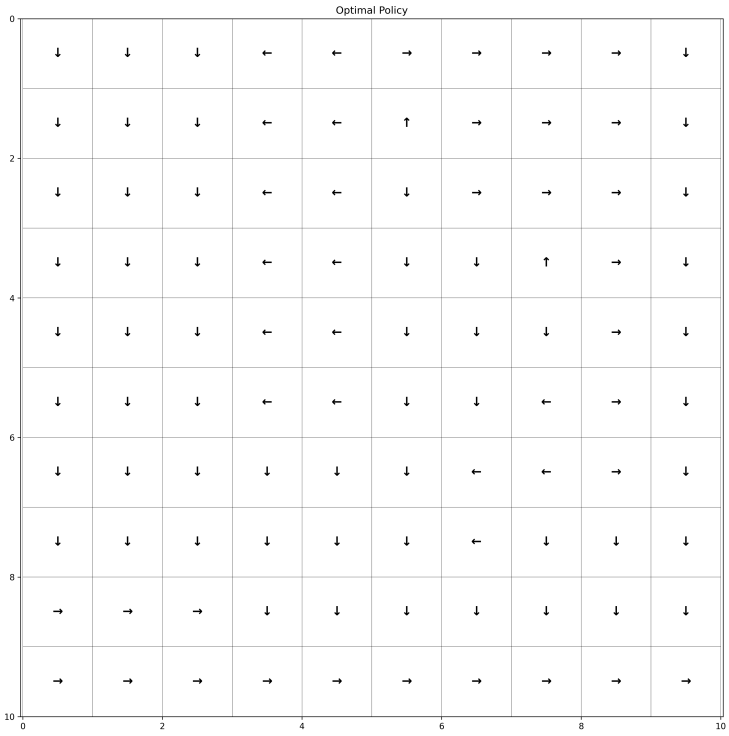


Figure 8: Optimal Action for Reward Function 2

From Figure 8, we observe several key points. Firstly, all arrows eventually direct the path towards state 99, which has the highest reward among all states. This is further supported by the heatmap in Figure 7, where arrows at each state point towards the state that will maximize the expected reward. The arrows also guide the agent away from regions with negative rewards (lighter areas on the heatmap) towards more red regions, aligning with the goal of maximizing cumulative rewards within a finite horizon. Therefore, the agent aims to reach state 99, as rewards from other states are either negative or zero.

Moreover, it is noticeable how the agent attempts to avoid the snake-like chain of negative rewards by moving towards regions that are spatially further from state 99. The agent's goal is not to reach the state with the highest reward in the shortest path, but rather to maximize the total cumulative reward within a finite horizon. This means the agent prefers a longer, more optimal path that

avoids negative rewards and ensures a higher overall reward.

Additionally, the agent can compute the optimal action to take at each state by observing the optimal values of its neighboring states. This is evident from the direction of arrows, which consistently point towards the neighboring state with the highest optimal value, as seen in Figure 6. Even if the agent does not know future optimal values, it can still develop an optimal policy by observing local optimal values. This ability allows the agent to make informed decisions based on the optimal values of neighboring states, leading to a strategy that maximizes expected rewards.

In conclusion, the agent's optimal policy depends on the optimal values of neighboring states, the transition matrix, and rewards. The value iteration algorithm guides the agent to choose actions that maximize the expected reward based on local observations. This approach ensures that the agent effectively navigates towards state 99 while avoiding negative rewards and maximizing cumulative rewards over time.

Question 9

The appropriate value for w is 0.1 as it increases the likelihood of the agent reaching the state with the highest reward. This value also allows for some exploration while minimizing the chances of the agent getting stuck in local optima or moving off the grid.

Question 10

We can get the equivalent LP using block matrices:

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Dx \leq b, \quad \forall a \in \mathcal{A} \setminus \{a_1\} \end{aligned}$$

where

$$c = \begin{bmatrix} \mathbf{1}_{|S|\times 1} \\ -\lambda \mathbf{1}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \end{bmatrix}, \quad x = \begin{bmatrix} t \\ u \\ R \end{bmatrix}$$

$$D = \begin{bmatrix} I_{|S|\times|S|} & 0 & (P_a - P_{a_1})(I - \gamma P_{a_1})^{-1} \\ 0 & 0 & (P_a - P_{a_1})(I - \gamma P_{a_1})^{-1} \\ 0 & -I_{|S|\times|S|} & I_{|S|\times|S|} \\ 0 & -I_{|S|\times|S|} & -I_{|S|\times|S|} \\ 0 & 0 & I_{|S|\times|S|} \\ 0 & 0 & -I_{|S|\times|S|} \end{bmatrix}$$

$$b = \begin{bmatrix} \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ (R_{\max})_{|S|\times 1} \\ (R_{\max})_{|S|\times 1} \end{bmatrix}$$

Question 11

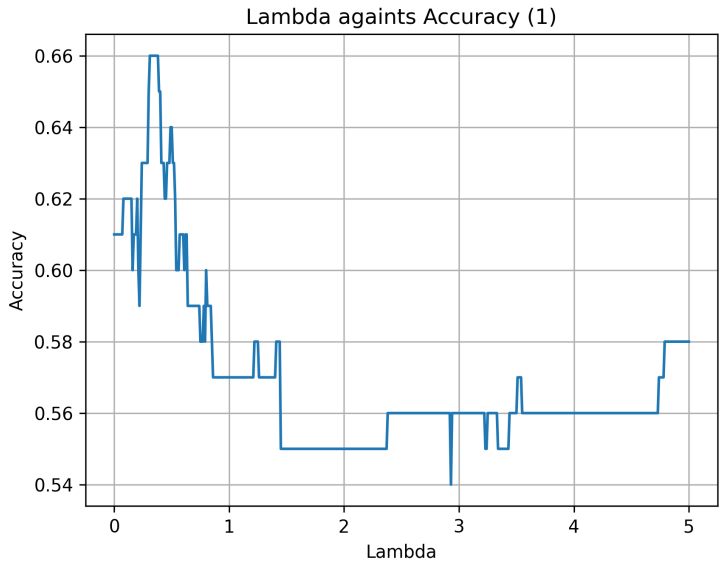


Figure 9: λ against Accuracy with Extracted Reward Function 1

Figure 9 shows that as λ increases, accuracy initially improves, achieving a globally optimal policy before declining. Subsequently, accuracy rises again but stabilizes at a suboptimal level. This pattern is expected because λ functions as a regularizer (similar to L1 or Lasso normalization), promoting simpler reward vectors. Limited values of λ enhance the robustness, clarity, and transferability of the extracted reward function, which generalizes well in deriving the optimal policy, explaining the initial accuracy boost. However, excessively high values of λ result in reward vectors that do not adequately address the tasks and lack the complexity needed for a global optimal policy, increasing the likelihood of getting trapped in local optima.

Question 12

$\lambda_{\max}^{(1)}$ is 0.31 and the maximum accuracy is 66%.

Question 13

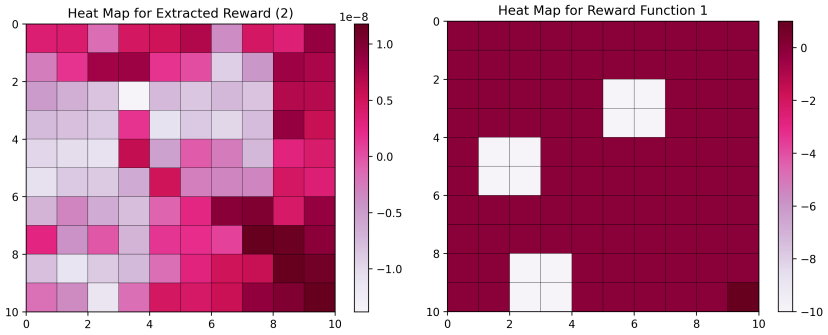


Figure 10: Heat Map for Extracted Reward and Ground Truth Reward (1)

Figure 10 illustrates that the IRL algorithm has effectively pinpointed areas around state 99 as high reward zones, while regions near negative-reward areas are assigned low immediate rewards. As we move away from state 99, which offers the highest reward, the immediate rewards decrease. Similarly, neighboring states near negative-reward blocks have lower extracted rewards compared to those near state 99. The extracted rewards are more continuous and dispersed across the state space than the actual reward function because they are derived from the expert policy rather than the true rewards. The expert policy reflects optimal values that change gradually, unlike the discrete nature of the reward function. This difference in scale between the extracted reward function and the expert reward function is due to the use of policy for reward extraction.

Learning from the policy equips the agent with better insights about the state-space in terms of rewards. As the agent approaches state 99, it observes higher rewards. This gradual increase or decrease in rewards near critical states (those with very high or low rewards) helps the agent develop better policies for dealing with rare rewards. The agent gains a clearer understanding of the state-space from the slowly changing immediate rewards at each state. This feedback offers the agent valuable heuristics for navigation, especially useful when the state-space is only partially observable and decisions must be made based on immediate rewards.

Question 14

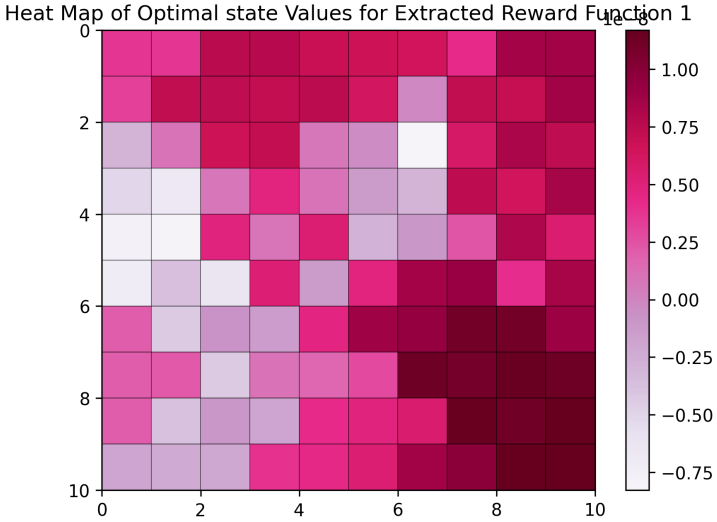


Figure 11: Heat Map for Optimal State Values for Extracted Reward Function with $\lambda_{max}^{(1)}$

Question 15

In comparing the heatmaps from Questions 3 and 14, both show that the area around state 99 has the highest optimal value due to its high reward. As one moves away from state 99, the values gradually decrease, and negative reward areas exhibit low optimal values in both plots. The patterns from the heatmap of reward function 1, including three blocks of negative rewards, are visible in both heatmaps. The gradual decay of optimal values around the highest reward state, due to the discount factor in the Bellman equation, is evident in both plots as well.

However, there are notable differences between the heatmaps. The heatmap in Question 14 has a smaller scale because it is derived from extracted rewards based on the expert policy, unlike the ground truth rewards used in Question 3. The regions in Question 3's heatmap are more defined and homogeneous, while those in Question 14 are more continuous and spread out due to being based on the expert policy. Additionally, the agent is less harshly penalized in low reward regions in Question 14 compared to Question 3, and there is a higher likelihood of the agent deviating from the optimal state (state 99) due to accumulating local rewards that may lead it astray. This results in the agent potentially achieving a lower expected reward and being more likely to move off the grid in Question 14's state-space.

Question 16

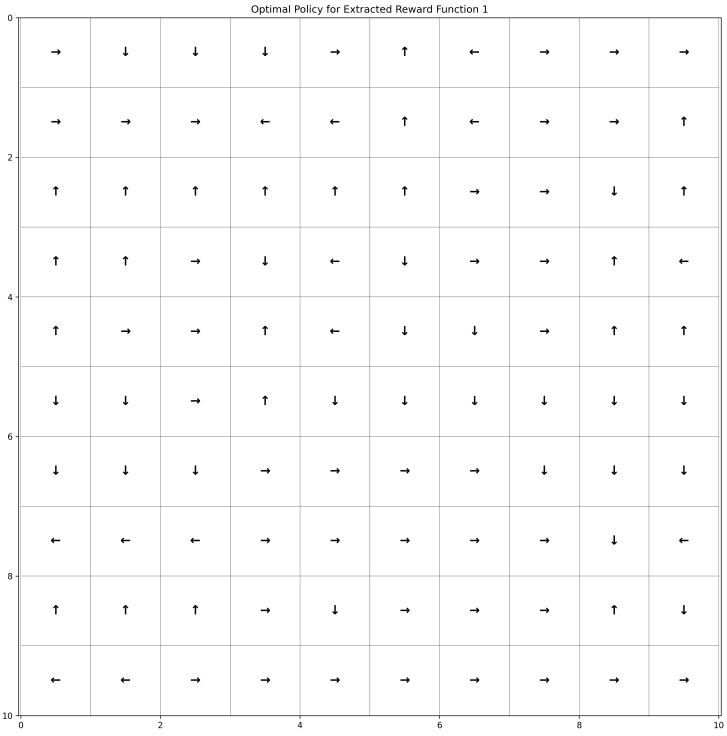


Figure 12: Optimal Actions for Extracted Reward Function with $\lambda_{max}^{(1)}$

Question 17

In both Questions 5 and 16, the majority of actions in the policy maps involve moving down and to the right, aiming to reach the most optimal state, state 99, located in the lower right corner of the state-space. This indicates that in both cases, the agent is directed towards states with the highest rewards. However, there are differences between the two scenarios. In Question 16, some actions can lead the agent off the grid, unlike in Question 5. This happens because there is a higher likelihood of the agent not moving towards state 99 in Question 14's state-space compared to Question 3's state-space. Immediate and local rewards might mislead the agent, causing it to accumulate a lower expected reward over time and increasing the risk of suboptimal actions or going off the grid.

Additionally, in Question 16, some actions cause the agent to oscillate between two states, creating a deadlock condition due to similar and non-zero state values between neighboring states. This deadlock is absent in Question 5, where most state rewards are zero, allowing better foresight and influence of high reward states. The policy map in Question 5 ensures that the agent reaches state 99 from any position, while in Question 16, the agent faces higher risks of being blown off the grid or getting stuck in local optima. Furthermore, there is more variance and unpredictability in the actions in Question 16, compared to the more ordered and consistent actions aimed at reaching state 99 in Question 5.

Question 18

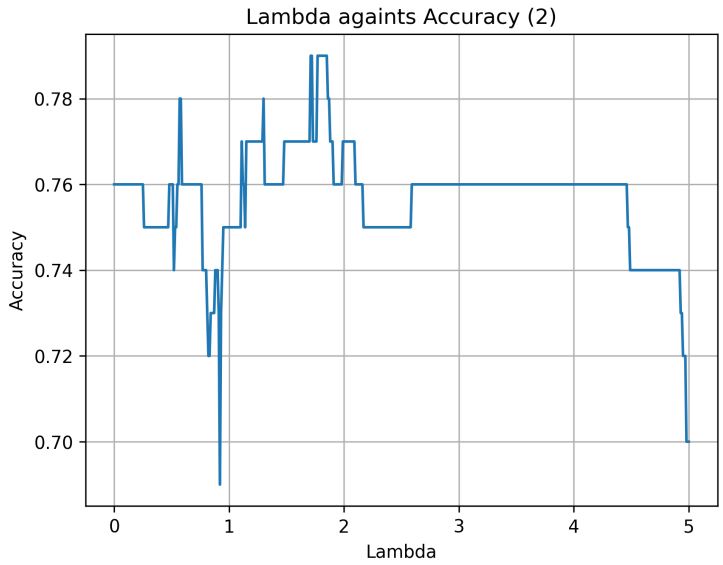


Figure 13: λ against Accuracy with Extracted Reward Function 1

Figure 13 demonstrates that as λ increases, accuracy initially declines until it reaches around 1, after which it begins to improve until it reaches approximately 2. At $\lambda = 2$, accuracy slightly drops and stabilizes, indicating a local optimum, until about 4.5, beyond which it declines again. This behavior is anticipated since λ serves as a regularizer (like L1 or Lasso normalization), promoting simpler reward vectors. Moderate values of λ enhance the robustness, clarity, and transferability of the extracted reward function, aiding in the generalization needed for extracting the optimal policy, which explains the initial rise in accuracy. However, excessively high values of λ result in reward vectors that are insufficiently complex for the tasks at hand, making it difficult to derive a globally optimal policy and increasing the risk of being trapped in a local optimum.

Question 19

$\lambda_{\max}^{(2)}$ is 1.71 and the maximum accuracy is 79%.

Question 20

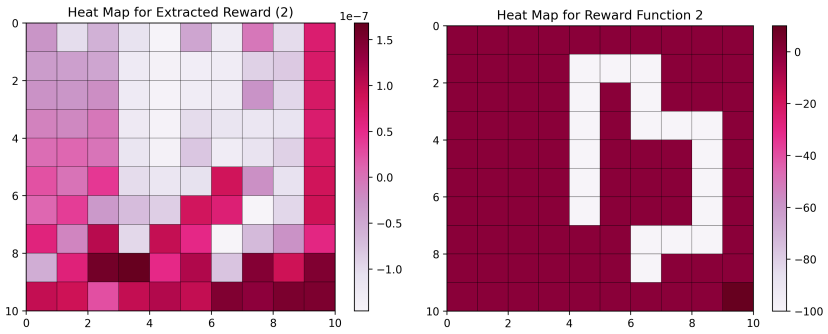


Figure 14: Heat Map for Extracted Reward and Ground Truth Reward (1)

From Figure 14, we observe that the rewards extracted by the IRL algorithm are more evenly distributed and continuous compared to the actual reward function. This difference arises because the rewards are derived from the expert policy rather than the true rewards. The expert policy reflects optimal values that change gradually, unlike the discrete nature of the actual reward function, leading to differences in scale between the extracted and true reward functions.

Furthermore, the IRL algorithm accurately identifies the high-reward region around state 99, while areas near the negative-reward chain have low immediate rewards. As one moves away from state 99, the immediate rewards decrease, and neighboring states near negative-reward areas also show lower rewards in the extracted reward heatmap. However, the IRL algorithm also identifies another high-reward region in the lower left corner (around states 28 and 38), which is counterintuitive since state 99 is expected to have the highest reward. This discrepancy occurs because the reward function alone is not a reliable indicator of long-term returns and should be used with the transition matrix and optimal state values. IRL, by learning from the expert policy, equips the agent with better heuristics for navigating the state-space. As the agent approaches state 99, it observes higher rewards, and this gradual change in rewards near critical states helps the agent develop better policies for rare rewards. This feedback is particularly useful when the state-space is only partially observable and decisions must be made based on immediate rewards.

Question 21

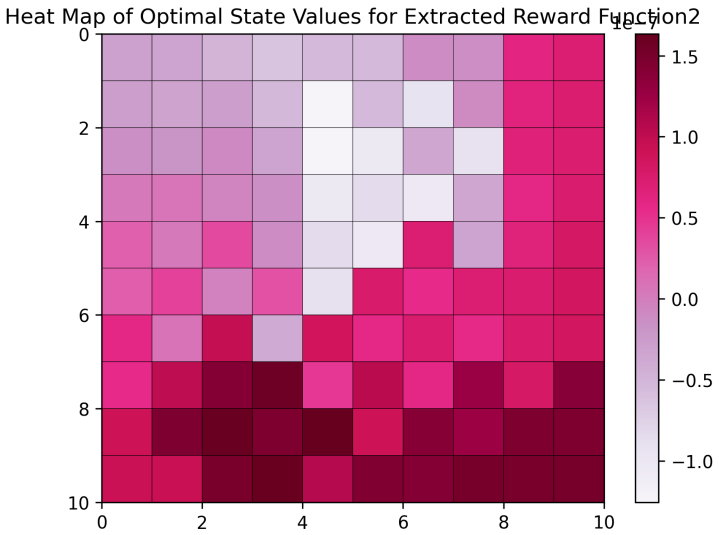


Figure 15: Heat Map for Optimal State Values for Extracted Reward Function with $\lambda_{max}^{(2)}$

Question 22

When comparing the heatmaps from Questions 7 and 21, both display similarities such as regions with negative rewards being similarly located and the area around state 99 having high optimal values, although not always the highest in both plots. As the agent moves away from state 99, values decrease in both cases. The gradual decline in optimal values as the agent moves towards states with lower rewards is evident in both heatmaps due to the Bellman equation's discount factor. This gradual change is more pronounced in the heatmap of Question 21.

However, notable differences exist between the two heatmaps. In Question 21, the highest optimal values shift to the lower-left corner, contrasting with the high values around state 99 in Question 7. This discrepancy is due to the reward function being derived from the expert policy in Question 21, leading to a more spread-out and continuous distribution compared to the discrete values in Question 7. The scaling in Question 21's heatmap is smaller, as it is based on extracted rewards. Regions in Question 7's heatmap are more defined and homogeneous, while Question 21's are more dispersed. Additionally, the agent in Question 21 faces less severe penalties near low-reward regions and has a higher likelihood of straying from the optimal path, resulting in a lower expected reward and increased chances of moving off the grid compared to Question 7.

Question 23

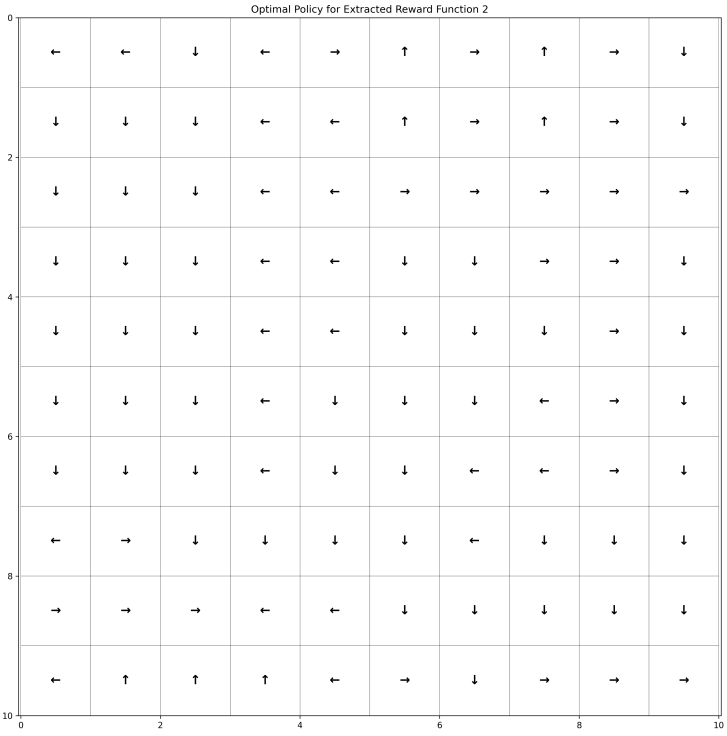


Figure 16: Optimal Actions for Extracted Reward Function with $\lambda_{max}^{(2)}$

Question 24

In both Questions 8 and 23, the agent moves away from low-reward or low-optimal value regions towards areas with high rewards or high optimal values. Both cases show some arrows diverging, possibly due to incompatible states, and the actions are evenly distributed in all four directions (up, down, left, and right), indicating that the agent explores various paths to reach optimal states.

However, there are differences between the two scenarios. Question 23 has two high-reward regions compared to one in Question 8, with the IRL algorithm identifying a high-reward area in the lower left corner (around states 28 and 38). Some actions in Question 23 can lead the agent off the grid, unlike in Question 8. This increased risk in Question 23 is due to local rewards potentially misleading the agent away from optimal paths, resulting in lower long-term rewards and higher chances of suboptimal actions. Additionally, some actions in Question 23 create a deadlock condition where the agent oscillates between two states, a situation not present in Question 8. The policy map in Question 8 ensures the agent reaches the optimal state from any position, whereas in Question 23, there is a higher risk of the agent being blown off the grid or getting stuck in local optima. There is also more variance and unpredictability in the actions in Question 23 compared to the more orderly actions in Question 8, which are aimed at reaching state 99.

Question 25

There are two discrepancies occurring in the action map in Question 23: moving off the grid and the deadlock condition. Moving off the grid occurs when some actions cause the agent to be blown off the grid. This is due to the high probability that the agent will move in the wrong direction, away from the optimal region, thereby overpowering the maximum achievable reward. In other words, the agent will accumulate a lower expected reward in the long run for the state space in Question 21 compared to the agent in Question 7. This ultimately causes the agent to be more likely to move off the grid or take suboptimal actions over time.

The deadlock condition occurs when an agent oscillates between two states indefinitely. This results from gradually changing immediate rewards among neighboring states. If the state values between two neighboring states are very similar and non-zero, the agent will be forced to go back and forth between the two states if the exploration probability is low. The figure below shows the occurrence of the two discrepancies.

To address the issue of the agent being blown off the grid, we hardcoded the values for the edge and corner states (excluding state 99) to $-\infty$. This adjustment forces the value iteration algorithm to select states within the grid, effectively limiting the possible set of actions in edge and corner states. Additionally, we reduced the value of ϵ from 0.01 to 10^{-10} to establish a stricter convergence criterion. This change causes the value iteration algorithm to take more epochs to converge, thereby increasing the likelihood of finding more optimal policies and improving the accuracy of the IRL algorithm. The resulting optimal policies for both expert reward functions (reward functions 1 and 2) are displayed in Figure 17.

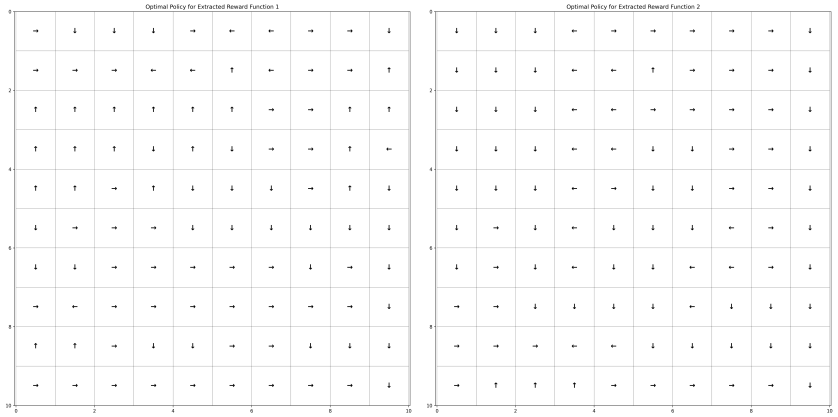


Figure 17: Heat Map for Extracted Reward and Ground Truth Reward (1)

Figure 17 demonstrates that the initial modification successfully keeps the agent within the grid in both instances, effectively preventing it from being blown off the grid. The agent's movements are entirely contained within the grid boundaries. Moreover, a greater number of paths now direct the agent towards the optimal state (state 99) in both cases.

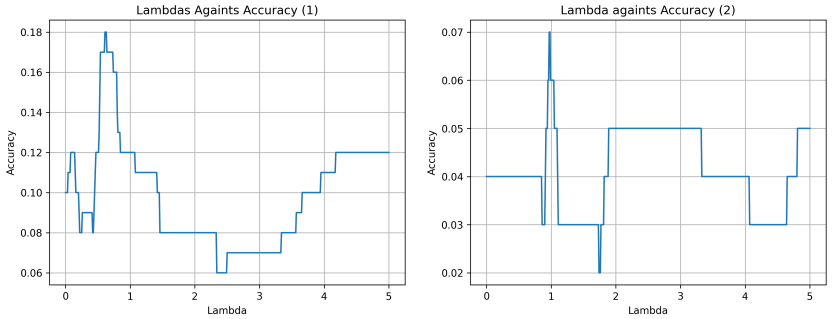


Figure 18: λ against Accuracy with Extracted Reward Function 1 and Extracted Reward Function 2

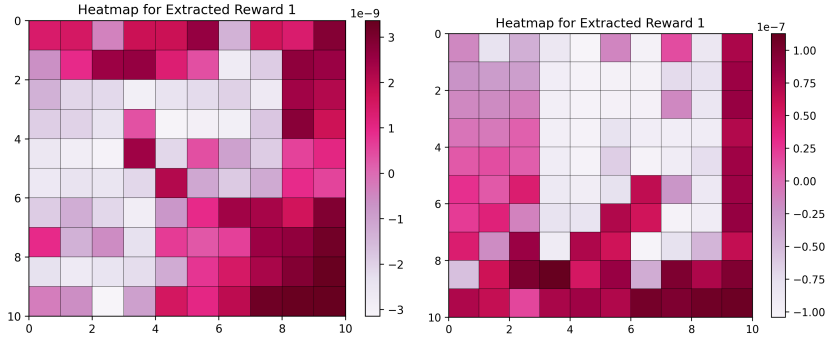


Figure 19: Heat Map for Extracted Reward with $\lambda^{(1)}_{max}$ and $\lambda^{(2)}_{max}$

For reward function the new $\lambda^{(1)}_{max}$ is 0.61 and the new $\lambda^{(2)}_{max}$ is 0.97.

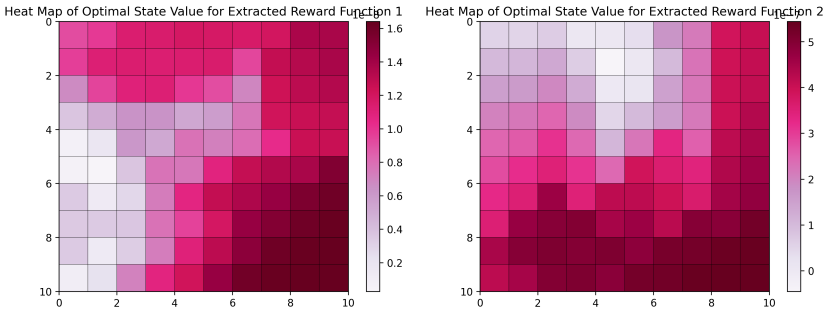


Figure 20: Heat Map for Optimal State Values for Extracted Reward Function with $\lambda_{max}^{(1)}$ and $\lambda_{max}^{(2)}$

Figures 19 and 20 reveal that the areas around the optimal states are now much more distinct from the negative reward regions. This contrasts with the previous policy where the agent was prone to moving off the grid and ϵ was high. This change has increased the likelihood of the agent reaching the optimal state (state 99). The heatmaps for both reward and optimal value now closely resemble those of the expert's reward function and optimal value heatmaps.