Assignement: ex2

Ian Steenstra

September 2024

1 Problem 1

- **(a)** $S = \{x, y\}$ where $x \in [0, 10]$, $y \in [0, 10]$, and $\{x, y\}$ is not a wall; $A = \{LEFT, RIGHT, UP, DOWN\}$
- **(b)** 45x4 + 41x3 + 20x2 = 343; where there are 45 states that can take 4 different actions, 41 states that can take 3 different actions, and 20 states that can only take 2 different actions. Taking into account the reward doesn't matter, as each state-action pair has only 1 reward with a non-zero possibility of returning. Wall states are also not taken into account.

2 Problem 2

- **(a)**
  - $G_5 = 0$
  - $G_4 = \gamma^0 R_5 = 2$
  - $G_3 = \gamma^0 R_4 + \gamma^1 G_4 = 4$
  - $G_2 = \gamma^0 R_3 + \gamma^1 G_3 = 8$
  - $G_1 = \gamma^0 R_2 + \gamma^1 G_2 = 6$
  - $G_0 = \gamma^0 R_1 + \gamma^1 G_1 = 2$
- **(b)**
  - $G_1 = \gamma^0 7 + \gamma^1 7 + ... = \frac{7}{1-0.9} = 70$
  - $G_0 = \gamma^0 R1 + \gamma^1 G_1 = 65$

3 Problem 3

- **(a)** Because either the episodic or continuous could go on for a long time, given a high enough K for the episodic system, both could be approximately 0, as any failure in a far future state will have a minimal effect because of the discounting weights. However, if failure occurs reasonably

soon, then $G_t = 0 + 0\gamma + 0\gamma^2 + \cdots + 0\gamma^{K-t-1} - 1\gamma^{K-t}; G_t = -\gamma^{K-t}$ would be returned for both episodic and continuous because the discounting factor γ already places less weight on rewards further into the future, making the primary concern the immediate possibility of failure. In both settings, the return at any time 't' before failure is the same: $-\gamma^{K-t}$.

- **(b)** The problem lies in the reward structure. Giving a reward only when the robot escapes the maze and zero otherwise doesn't incentivize it to escape quickly. The robot receives the same total reward (+1) whether it takes 10 steps or 1 million steps. This means the robot has no motivation to improve its path-finding strategy. It might find the goal eventually, but it won't learn to do so efficiently. We have not effectively communicated the desire for speed to the agent.

  To fix this, I would provide a small negative reward for every time step the robot is in the maze. This encourages it to reach the goal faster to maximize the total reward.

4 Problem 4

- **(a)** The signs of the rewards (+10, +5, -1, 0) help define the goal and penalties, but the agent's learning and decision-making are primarily driven by the differences in value between states or the intervals. The reason can be shown by adding a constant c to all rewards:

  Let's start with the Bellman equation for v_π (Equation 3.12), but with v'_π instead:

  $v'_\pi(s) = \sum_a \pi(a|s) \sum_{s',r'} p(s', r'|s, a)[r' + \gamma v'_\pi(s')]$

  Then add c to each new reward: $r' = r + c$

  $v'_\pi(s) = \sum_a \pi(a|s) \sum_{s',(r+c)} p(s', (r + c)|s, a)[(r + c) + \gamma v'_\pi(s')]$

  Then, break apart:

  $v'_\pi(s) = \sum_a \pi(a|s) \sum_{s',(r+c)} p(s', (r+c)|s, a)[(r+\gamma v'_\pi(s')] + c \sum_{s',(r+c)} p(s', (r+c)|s, a)$

  Now, substitute 1 for all summations that equal 1:

  $v'_\pi(s) = \sum_{s',(r+c)} p(s', (r + c)|s, a)[(r + \gamma v'_\pi(s')] + c$

  Replace with Equation 3.8:

  $v'_\pi(s) = v'_\pi[(r + \gamma v'_\pi(s')] + c$

  If we assume $v'_\pi = v_\pi + v_c$ and simplify:

  $v_\pi + v_c = v_\pi + \gamma v_c + c$

  And then get: $v_c = \frac{c}{1-\gamma}$

  Thus, showing that adding c provides a constant value.

- **(b)** Adding a constant c would have an effect on an episodic task because all rewards will shift the values of all states, including the terminal state. This change in the terminal state's value will have an effect on the value function, altering the relative values between states and potentially changing the optimal policy.

  Let's say we have a maze with three states: Beginning (B), Middle (M), and End (E). Rewards: $BtoM = 0, MtoE = +1, \&Bto* = 0$. Also, $\gamma = 0.9$

  $v(E) = 0; v(M) = 1 + 0.9v(E) = 1; v(B) = 0 + 0.9v(M) = 0.9$

  If we now add 1 to each reward, we would get:

  $v(E) = 1; v(M) = 2 + 0.9v(E) = 2.9; v(B) = 1 + 0.9v(M) = 3.61$

  The reward differences would result in different behavior of the agent.

5 Problem 5

- **(a)** $v_\pi(\text{center}) = \frac{1}{4}[0+0.9v_\pi(\text{up})]+\frac{1}{4}[0+0.9v_\pi(\text{right})]+\frac{1}{4}[0+0.9v_\pi(\text{down})]+\frac{1}{4}[0+0.9v_\pi(\text{left})]$

  $0.7 = \frac{1}{4}[0 + 0.9 \cdot 2.3] + \frac{1}{4}[0 + 0.9 \cdot 0.4] + \frac{1}{4}[0 + 0.9 \cdot -0.4] + \frac{1}{4}[0 + 0.9 \cdot 0.7]$

  $0.7 = \frac{1}{4}[2.07] + \frac{1}{4}[0.36] + \frac{1}{4}[-0.36] + \frac{1}{4}[0.63]$

  $0.7 = 0.5175 + 0.09 - 0.09 + 0.1575$

  $0.7 = 0.675$ (rounded to 0.7)

- **(b)** Since two actions are equally optimal and the others are not, I'll verify the Bellman equation for both optimal actions:

  - Action: UP (↑)
    Next state (s'): State above the center with $v^*(s') = 19.8$

    $$\begin{aligned} v^*(\text{center}) &= R(\text{center}, \text{UP}) + \gamma v^*(\text{UP}) \\ 17.8 &= 0 + 0.9 \times 19.8 \\ 17.8 &= 17.82 \text{ (rounded to 17.8)} \end{aligned}$$

  - Action: LEFT (←)
    Next state (s'): State to the left of the center with $v^*(s') = 19.8$

    $$\begin{aligned} v^*(\text{center}) &= R(\text{center}, \text{LEFT}) + \gamma v^*(\text{LEFT}) \\ 17.8 &= 0 + 0.9 \times 19.8 \\ 17.8 &= 17.82 \text{ (rounded to 17.8)} \end{aligned}$$

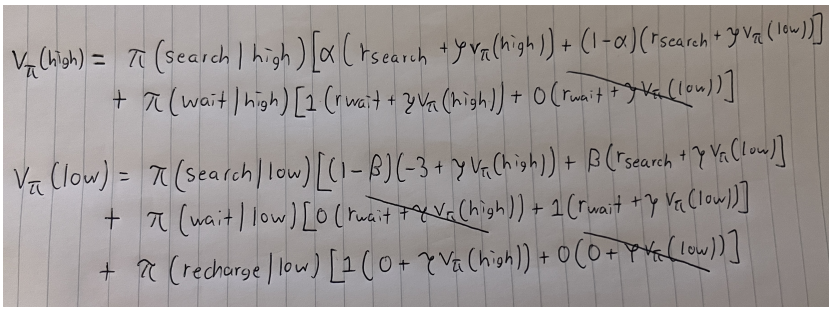

Figure 1: Problem 6(a): State-Value Functions for Policy

6 Problem 6

- **(a)** Using Table 3.1, see Figure 1.
- **(b)** Given the parameters $\alpha = 0.8, \beta = 0.6, \gamma = 0.9, r_{search} = 10, r_{wait} = 3$, and the policy $\pi(search|high) = 1, \pi(wait|low) = 0.5, \pi(recharge|low) = 0.5$, the value function for this policy is (see Figure 2):

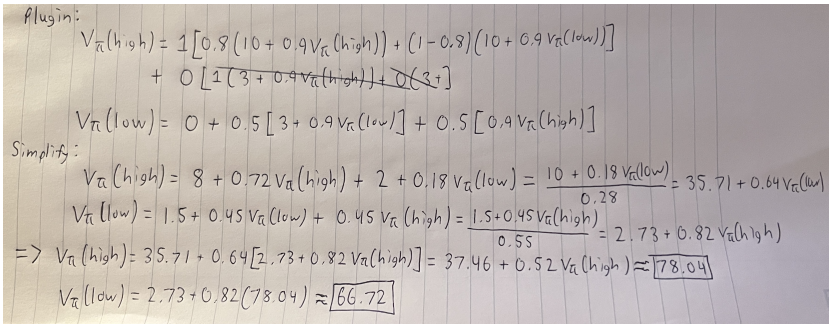

Figure 2: Problem 6(b): Plugged In State-Value Functions for Policy

This means that, under the given policy and parameters, the long-term expected discounted reward is approximately 78.04 when starting in the 'high' battery state and approximately 66.72 when starting in the 'low' battery state.