1 Problem 1

Problem1.1

Value Function:

$$E[\sum_{t=0}^{\tau-1} \alpha^t r(s_t, a_t)|s_0] \tag{1}$$

Optimal Value Function:

$$V^*(s) = \max_{\mu} E[\sum_{t=0}^{\tau-1} \alpha^t r(s_t, \mu(s_t))|s_0] \tag{2}$$

Bellman Equation:

$$V_t^*(s_t) = \begin{cases} 0 & s = 6 \\ \max_a E[10 + 0.9 * V_{t+1}(f(s_t, a_t))] & s = 5, 7, 14 \\ \max_a E[-1.5 + 0.9 * V_{t+1}(f(s_t, a_t))] & s \neq 6, 5, 7, 14 \end{cases}$$

Problem1.2

$$V_0 = \begin{bmatrix} v_0 = 0 \\ v_1 = 0 \\ v_2 = 0 \\ \vdots \\ v_{63} = 0 \end{bmatrix} \tag{3}$$

If agent is on state 6, then terminate.

$$V_1(6) = 0 \tag{4}$$

If agent is on state 5, 7, or 14, the policy is always to move toward state 6, so the reward is 10:

$$V_1(5) = 10 \tag{5}$$

$$V_1(7) = 10 \tag{6}$$

$$V_1(14) = 10 \tag{7}$$

If the agent is on any other state, there will be no way to access state 6, so the equation to calculate the optimal value is:

$$V_1(s) = \max_a(-1.5 + 0.9 * V_0(f(s_0, a_0))) \tag{8}$$

Consdidering all states adjecent to state 5, 7, and 14:

$$V_1(4) = \max([-1.5+0.9*V_0(f(4,0)), -1.5+0.9*V_0(f(4,1)), -1.5+0.9*V_0(f(4,2)), -1.5+0.9*V_0(f(4,3))]) \tag{9}$$

$$V_1(4) = \max([-1.5+0.9*V_0(4), -1.5+0.9*V_0(5), -1.5+0.9*V_0(12), -1.5+0.9*V_0(3)]) \tag{10}$$

$$V_1(4) = \max([-1.5+0.9*0, -1.5+0.9*10, -1.5+0.9*0, -1.5+0.9*0]) \tag{11}$$

$$V_1(4) = \max([-1.5+0, -1.5+9, -1.5+0, -1.5+0]) \tag{12}$$

$$V_1(4) = \max([-1.5, 7.5, -1.5, -1.5]) \tag{13}$$

$$V_1(4) = 7.5 \tag{14}$$

$$V_1(13) = 7.5 \tag{15}$$

$$V_1(15) = 7.5 \tag{16}$$

$$V_1 = \begin{bmatrix} v_0 = -1.5 \\ v_1 = -1.5 \\ v_2 = -1.5 \\ v_3 = -1.5 \\ v_4 = 7.5 \\ v_5 = 10 \\ v_6 = 0 \\ v_7 = 10 \\ v_8 = -1.5 \\ \vdots \\ v_{12} = -1.5 \\ v_{13} = 7.5 \\ v_{14} = 10 \\ v_{15} = 7.5 \\ v_{16} = -1.5 \\ \vdots \\ v_{62} = -1.5 \\ v_{63} = -1.5 \end{bmatrix} \tag{17}$$

Problem1.3

Assume initial policy $\mu_0(s) = 0$ for any $s \in \{0, 1, \dots, 63\}$
Beside state 6 and state 14(that is directly below state 6), the reward will always be -1.5.

The value vector for the initial policy V_{μ_0} is:

$$V_{\mu_0} = \begin{bmatrix} v_0 = -1.5 \\ v_1 = -1.5 \\ v_2 = -1.5 \\ v_3 = -1.5 \\ v_4 = -1.5 \\ v_5 = -1.5 \\ v_6 = 0 \\ v_7 = -1.5 \\ \vdots \\ v_{13} = -1.5 \\ v_{14} = 10 \\ v_{15} = -1.5 \\ \vdots \\ v_{62} = -1.5 \\ v_{63} = -1.5 \end{bmatrix} \tag{18}$$

The optimal policy μ_1 is:

$$\mu_1 = \begin{bmatrix} \mu_1(0) = 0 \\ \mu_1(1) = 0 \\ \mu_1(2) = 0 \\ \mu_1(3) = 0 \\ \mu_1(4) = 0 \\ \mu_1(5) = 0 \\ \mu_1(6) = 0 \\ \mu_1(7) = 0 \\ \vdots \\ \mu_1(12) = 0 \\ \mu_1(13) = 1 \\ \mu_1(14) = 0 \\ \mu_1(15) = 3 \\ \mu_1(16) = 0 \\ \vdots \\ \mu_1(62) = 0 \\ \mu_1(63) = 0 \end{bmatrix} \tag{19}$$

2 Problem 2

Consider a Markov chain with three states $\{1, 2, 3\}$. In each state, we can choose one of the two possible actions $\{1, 2\}$. The transition probability matrices under the two actions are given below:

$$P(1) = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.1 & 0.4 & 0.5 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} P(2) = \begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.5 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.3 \end{pmatrix} \tag{20}$$

The cost for a given (state, action) pair is a Bernoulli random variable. The mean costs are given below

$$C = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.1 \\ 0 & 0 \end{pmatrix} \tag{21}$$

We are interested in solving the following discounted cost problem

$$\min_{\mu} \lim_{N\to\infty} E[\sum_{k=0}^{N} 0.9^k c(x_k, u_k)|x_0 = 1, u_0 = 1] \tag{22}$$

where x_k is the state at time k, u_k is the action at time k, and μ denotes a policy. Assume we do not know the model but are given the following trace $(x_k, u_k, c(x_k, u_k))$ instead:

$$(1, 1, 0) \to (2, 1, 1) \to (3, 2, 0) \to (2, 2, 1) \tag{23}$$

Consider the Q-learning algorithm with $Q_0 = \begin{pmatrix} 0 & 0.5 \\ 0.3 & 0 \\ 0.2 & 0.1 \end{pmatrix}$ and step size $\epsilon = 0.1$. Please calculate the sequence of Q-values under Q-learning with the trace given above.

Q-Value is calculated using the following equation:

$$Q_{k+1}(x_k, u_k) = Q_k(x_k, u_k) + \beta(c(x_k, u_k) + \alpha \min_v Q_0(x'_k, v) - Q_k(x_k, u_k)) \tag{24}$$

$$Q_1(1, 1) = Q_0(1, 1) + \beta(c(1, 1) + \alpha \min_v Q_0(2, v) - Q_0(1, 1)) \tag{25}$$

$$Q_1(1, 1) = Q_0(1, 1) + 0.1 * (c(1, 1) + 0.9 * \min_v Q_0(2, v) - Q_0(1, 1)) \tag{26}$$

$$Q_1(1, 1) = 0 + 0.1 * (0 + 0.9 * \min_v[Q_0(2, 1), Q_0(2, 2)] - 0) \tag{27}$$

$$Q_1(1,1) = 0 + 0.1 * (0 + 0.9 * \min[0.3, 0] - 0) \tag{28}$$

$$Q_1(1,1) = 0 \tag{29}$$

The new Q-learning algorithm is

$$Q_1 = \begin{pmatrix} 0 & 0.5 \\ 0.3 & 0 \\ 0.2 & 0.1 \end{pmatrix} \tag{30}$$

Continue the process for the rest of the trace.

$$Q_2(2,1) = Q_1(2,1) + \beta(c(2,1) + \alpha \min_v Q_1(3,v) - Q_1(2,1)) \tag{31}$$

$$Q_2(2,1) = 0.3 + 0.1 * (1 + 0.9 * \min[0.2, 0.1] - 0.3) \tag{32}$$

$$Q_2(2,1) = 0.379 \tag{33}$$

$$Q_2 = \begin{pmatrix} 0 & 0.5 \\ 0.379 & 0 \\ 0.2 & 0.1 \end{pmatrix} \tag{34}$$

$$Q_3(3,2) = Q_2(3,2) + \beta(c(3,2) + \alpha * \min_v Q_2(2,v) - Q_2(3,2)) \tag{35}$$

$$Q_3(3,2) = 0.1 + 0.1 * (0 + 0.9 * \min[0.379, 0] - 0.1) \tag{36}$$

$$Q_3(3,2) = 0.09 \tag{37}$$

$$Q_2 = \begin{pmatrix} 0 & 0.5 \\ 0.379 & 0 \\ 0.2 & 0.09 \end{pmatrix} \tag{38}$$