1.

(a) $V_*(s) = \max_a q_*(s,a)$

(b) $q_*(s,a) = \sum_{s',r} p(s',r|s,a)[r+\gamma V_*(s')]$

(c) $\pi_*(s) = \arg\max_a q_*(s,a)$

(d) $\pi_*(s) = \arg\max_a \{ \sum_{s',r} p(s',r|s,a)[r+\gamma V_*(s')] \}$

(e) $p(s'|s,a) = \sum_r p(s',r|s,a)$

$r(s,a) = \sum_r r \sum_{s'} p(s',r|s,a)$ $\qquad$ $V_\pi = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)(r+\gamma V_\pi(s'))$

Hence,

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

$$V_*(s) = \max_a [r(s,a) + \sum_{s'} p(s'|s,a)\cdot \gamma V_\pi(s')]$$

$$q_\pi(s,a) = r(s,a) + \sum_{s'} p(s'|s,a)\cdot \gamma V_\pi(s')$$

$$q_*(s,a) = r(s,a) + \sum_{s'} p(s'|s,a)\cdot \gamma \max_{a'} q_*(s',a')$$

2.

(a) If two policies is equally good but taking different actions, it will never converge because of "If old-action $\neq \pi(s)$, then policy-state $\leftarrow$ false."

we need to change it to:

{ If $q(s, \text{old\_action}) \neq q(s, \pi(s))$, then: policy-state $\leftarrow$ false }

So make sure it realizes that these are two equally good policy.

(b) No. Value-Iteration calculate $\pi_*(s)$ by given V_*, instead of comparing the actions between two policies.

3.

(a) 1. Initialization:

$Q(s,a) \in \mathbb{R}$ arbitrarily for $s \in S$, $a \in A$

2. Policy Evaluation:

Loop: $\Delta \leftarrow 0$

Loop for each $s \in S$, $a \in A$

$q(s,a) \leftarrow Q(s,a)$

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

$\Delta \leftarrow \max(\Delta, |q(s,a) - Q(s,a)|)$

until $\Delta < \theta$

3. Policy Improvement

policy-stable $\leftarrow$ true

For each $s \in S$:

old-action $\leftarrow \pi(s)$

$\pi(s) = \arg\max_a Q(s,a)$

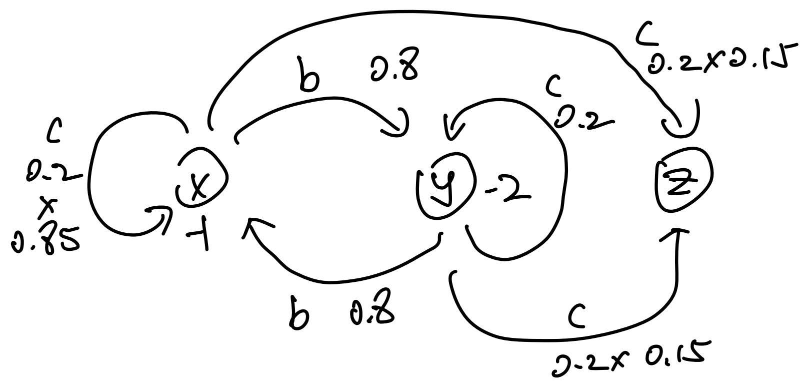
If old-action $\neq \pi(s)$, then policy-stable $\leftarrow$ false.

If policy-stable, then stop and return $Q \approx q_*$, and $\pi \approx \pi_*$

else go to Step 2.

(b) $$q_{k+1}(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} q(s',a')]$$

4.



(a) Since S_z (state Z) is a termial state and each step that is spent in S_x and S_y will pay a cost, the agent wants to get S_z as soon as possible. But only action c (A_c) can arrive at S_z with low posibility, the agent will only take A_c when it is at S_x as paying less cost. When it is at S_y, it is suggested that try A_b to get to S_x, where has less penalty, rather than take A_c directly for S_z.

(b)
$$V(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$$
$$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$$

Initialization: $V_0(s) \leftarrow (S_x, S_y, S_z)$, $\pi_0 \leftarrow (a_c, a_c)$

$$\begin{cases} V_1(S_x) = 0.85 \times [-1+V(S_x)] + 0.15 \times [-1+V(S_z)] \\ V_1(S_y) = 0.85 \times [-2+V(S_y)] + 0.15 \times [-2+V(S_z)] \\ V_1(S_z) = 0 \end{cases}$$

Solved $\rightarrow$
$$\begin{cases} V_1(S_x) = -6.67 \\ V_1(S_y) = -13.33 \\ V_1(S_z) = 0 \end{cases}$$

Policy Improvement 1

If it take a_b,

$V_b(S_x) = 0.8 \times (-1-13.33) + 0.2 \times (-1-6.67) = -13$

$V_b(S_y) = 0.8 \times (-2-6.67) + 0.2 \times (-2-13.33) = -10$

$$\begin{cases} V_b(S_x) < V_1(S_x) \\ V_b(S_y) > V_1(S_y) \end{cases} \quad \text{So} \quad \begin{cases} \pi_1(S_x) = a_c \\ \pi_1(S_y) = a_b \end{cases}$$ for Policy Improvement 1.

Policy Evaluation 1

$$\begin{cases} V_1(S_x) = 0.85 \times [-1+V_1(S_x)] + 0.15 \times [-1+V(S_z)] = -6.67 \\ V_1(S_y) = 0.2 \times [-2+V_1(S_y)] + 0.8 \times [-2+V_1(S_x)] = -9.34 \end{cases}$$

Policy Improvement 2.

If it takes a_b in S_x

$$V_b(S_x) = 0.8 \times (-1-9.34) + 0.2 \times (-1-6.67) = -9.81 < V_1(S_x)$$

If it takes a_c in S_y

$$V_c(S_y) = 0.85 \times (-2-6.67) + 0.15 \times (-2+0) = -7.67 < V_1(S_y)$$

So $\begin{cases} \pi_2(S_x) = a_c \\ \pi_2(S_y) = a_b \end{cases}$, $\pi(s)$ stay the same as previous one.

Terminate policy iteration

(c) If the inital policy has a_b in both states, then:

$$\begin{cases} V_1(S_x) = 0.8 \times [-1 + V(S_y)] + 0.2 \times [-1 + V(S_x)] \\ V_1(S_y) = 0.8 \times [-2 + V(S_x)] + 0.2 \times [-2 + V(S_y)] \\ V_1(S_z) = 0 \end{cases}$$

However, the fomula is unsolvable. Discounting will make it become solvable. The optimal policy depends on the discount factor. Assuming γ very small, the cost in the distant future makes less effect because $\gamma^n \approx 0$. Therefore, the agent might take action c, aiming directly to state z, regradless of the long-term effect by paying more cost.

6. (b) Change in _calculate_cost () function:

$\begin{cases} \text{One car can be moved from 1st location to 2nd location for free.} \\ \text{If state [0] > 10, then cost + 4} \\ \text{If state [1] > 10, then cost + 4.} \end{cases}$

The difference after the changes:

It becomes a non-linear problem which make the plots change.

We can see the policy plots are seperated by the lines at locA = 10 and locB = 10. That means they don't need to move the car when there are around 10 cars at both places.

But they need to move car when it is closed to 10 to avoid penalty.

7. (a) ① When $\max_a f(a) - \max_a g(a) \geq 0$,

$$\left|\max_a f(a) - \max_a g(a)\right| = \max_a f(a) - \max_a g(a) \leq \max_a f(a) - g(x) \quad \text{for } x \in R$$

Say that $a_1 = \arg\max_a |f(a) - g(a)|$, $a_2 = \arg\max_a f(a)$

$$f(a_1) - g(a_1) \geq f(a_2) - g(a_2)$$

Hence,

$$\max|f(a) - g(a)| \geq \max f(a) - g(a) \geq \max_a f(a) - \max_a g(a)$$

$$\max|f(a) - g(a)| \geq \max_a f(a) - \max_a g(a)$$

② When $\max_a f(a) - \max_a g(a) < 0$

$$\left|\max_a f(a) - \max_a g(a)\right| = \max_a g(a) - \max_a f(a) \leq \max_a g(a) - f(x) \quad \text{for } x \in R$$

For any a, we have $\max|g(a) - f(a)| \geq \max g(a) - f(a)$

So, $\max|g(a) - f(a)| \geq \max g(a) - f(a) \geq \max_a g(a) - \max_a f(a)$

Hence. $\left|\max_a f(a) - \max_a g(a)\right| \leq \max|f(a) - g(a)|$

(b) $$\| BV_i - BV_i' \| = \| \max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V_i(s')] - \max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V_i'(s')] \|$$

As we got $\left|\max_a f(a) - \max_a g(a)\right| \le \max |f(a) - g(a)|$,

$$\begin{aligned}
\| BV_i - BV_i' \| &\le \| \max_a \sum_{s',r} p(s',r|s,a)[\gamma (V_i(s') - V_i'(s'))] \| \\
&\le \max \left\{ \sum_{s',r} p(s',r|s,a)[\gamma (V_i(s') - V_i'(s'))], \dots \right\} \\
&\le \gamma \max \left\{ |V_i - V_i'|, |V_{i+1} - V_{i+1}'|, \dots \right\} \\
&\le \gamma \| V_i - V_i' \|_\infty
\end{aligned}$$