Zhuohang Li(zl299)

CS 520 Final: Question 2 - Markov Decision Processes

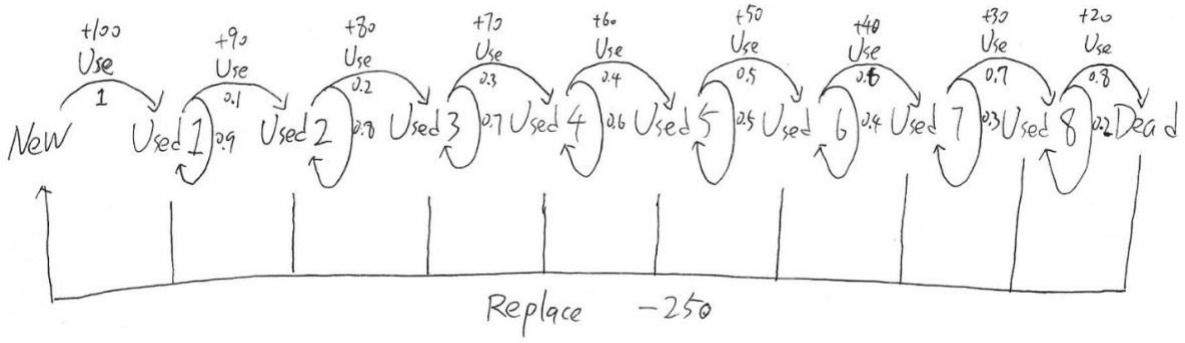
a) The process can be modeled as a Markov Decision Process. $P(s'|s,a)$is the probability of transiting to state s' when taking action a at state s, given by the following transition matrix:

transitions1

CurrentState	Action	NextState	Transition Prob
New	Use	Used1	1
Used1	Replace	New	1
Used1	Use	Used1	0.9
Used1	Use	Used2	0.1
Used2	Replace	New	1
Used2	Use	Used2	0.8
Used2	Use	Used3	0.2
Used3	Replace	New	1
Used3	Use	Used3	0.7
Used3	Use	Used4	0.3
Used4	Replace	New	1
Used4	Use	Used4	0.6
Used4	Use	Used5	0.4
Used5	Replace	New	1
Used5	Use	Used5	0.5
Used5	Use	Used6	0.5
Used6	Replace	New	1
Used6	Use	Used6	0.4
Used6	Use	Used7	0.6
Used7	Replace	New	1
Used7	Use	Used7	0.3
Used7	Use	Used8	0.7
Used8	Replace	New	1
Used8	Use	Used8	0.2
Used8	Use	Dead	0.8
Dead	Replace	New	1

rewards1

CurrentState	Action	Reward
New	Use	100
Used1	Replace	-250
Used1	Use	90
Used2	Replace	-250
Used2	Use	80
Used3	Replace	-250
Used3	Use	70
Used4	Replace	-250
Used4	Use	60
Used5	Replace	-250
Used5	Use	50
Used6	Replace	-250
Used6	Use	40
Used7	Replace	-250
Used7	Use	30
Used8	Replace	-250
Used8	Use	20
Dead	Replace	-250



$R(s,a)$ represents the reward of taking action a at the state of s, given by the reward matrix above.

Use value-iteration to solve the process and update the value function according to Bellman equation:

$$V'(s) = \max_a \sum_{s'} P(s'|s,a)(R(s,a) + \beta V(s'))$$

where β is the discount factor.
Iterate until:

$$\delta < \epsilon(1-\beta)/\beta$$

where δ is the maximum change in the utility of any state in an iteration, and ϵ is the maximum error allowed in the utility of any state.

The result is:

State	Value
New	800.530549926243
Used1	778.3673954680372
Used2	643.221234821306
Used3	556.1225096942012
Used4	502.8349428957192
Used5	475.84494368616834
Used6	470.47738893862834
Used7	470.47738893862834
Used8	470.47738893862834
Dead	470.47738893862834

b) The expected utility of taking action a at state s is given by the action value function:

$$Q(s,a) = R(s,a) + \beta \sum_{s'} P(s'|s,a)V(s')$$

Therefore, the best policy of state *s* is to perform the action with the maximum expected utility. The result is:

```
Optimal policy:
State   Action
New      Use
Used1    Use
Used2    Use
Used3    Use
Used4    Use
Used5    Use
Used6    Replace
Used7    Replace
Used8    Replace
Dead     Replace
```

c) To solve this problem, I made "BuyUsed" a new action available from state "Used2" to "Dead" (since it makes no sense to buy a used part in state "Used1"), with 0.5 probability to take the bot to state "Used1" and 0.5 to state "Used2". Also, in the reward table I mark the action "BuyUsed" with some amount of cost. And gradually reduce the cost of "BuyUsed" from 250. I found that with cost 169, the bot will choose "BuyUsed" over "Replace", but with cost 170, the bot will take "Replace" instead of "BuyUsed", which means the highest rational price for this option is 169.

d) Optimal policy for beta= 0.1:

State	Action
New	Use
Used1	Use
Used2	Use
Used3	Use
Used4	Use
Used5	Use
Used6	Use
Used7	Use
Used8	Use
Dead	Replace

Optimal policy for beta= 0.3:

State	Action
New	Use
Used1	Use
Used2	Use

Used3	Use
Used4	Use
Used5	Use
Used6	Use
Used7	Use
Used8	Use
Dead	Replace

Optimal policy for beta= 0.5:

State	Action
New	Use
Used1	Use
Used2	Use
Used3	Use
Used4	Use
Used5	Use
Used6	Use
Used7	Use
Used8	Use
Dead	Replace

Optimal policy for beta= 0.7:

State	Action
New	Use
Used1	Use
Used2	Use
Used3	Use
Used4	Use
Used5	Use
Used6	Use
Used7	Use
Used8	Use
Dead	Replace

Optimal policy for beta= 0.9:

State	Action
New	Use
Used1	Use
Used2	Use
Used3	Use

Used4	Use
Used5	Use
Used6	Replace
Used7	Replace
Used8	Replace
Dead	Replace

Optimal policy for beta= 0.99:

State	Action
New	Use
Used1	Use
Used2	Use
Used3	Use
Used4	Replace
Used5	Replace
Used6	Replace
Used7	Replace
Used8	Replace
Dead	Replace

According to the result, there is no optimal solution for all sufficiently large β. For smaller β values, i.e. $\beta = 0.1, 0.3, 0.5, 0.7$, the bot would only take REPLACE at state DEAD. This is with smaller β, the bot cares little about long-term gains, as long as it can get reward for taking USE, it will keep doing that. On the contrary, for larger β like 0.99, the bot will highly value the long-term gain. The bot will REPLACE at state "Used4" without hesitate, because comparing to take action USE at state "Used4" and get a reward of 60, it would rather transit to the state "New" and take USE to get a reward of 100, despite of the cost and actions that takes for it to get there.