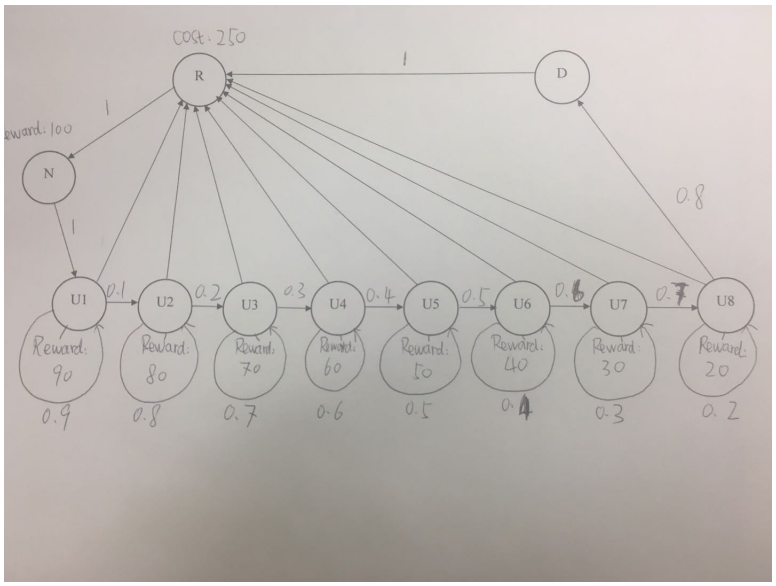
CS 520 Final Question 2 Solution

Guo Chen

December 2018

1 Solutions

The entire Markov state transition graph of this question is shown as below.



In this graph, "D" means the state "Dead", "R" means the state "Replace", and "N" means the state "New". All the transition probabilities are noted along with corresponding arrows. The reward written aside each state circle means the reward obtained as the flow approaches the state each time, no matter it comes from other states or itself. And since replacement can occur in any state, all the utility states have arrows pointing to the state "Replace".

To answer questions for question a) to d), I write a program which can run Markov decision process to calculate corresponding variables that are required. Specifically, as shown in the material, we can define the ***infinite horizon expected discounted utility*** starting from state s under policy π as $U_\pi(s) = E[\sum_{t=0}^{\infty} \beta^t r s_{t,\pi(S_t)} | S = S_0]$, which can be regarded as the limitation of $U_\pi(s)$, if it converges to a certain value. In addition, as in the material, all states

transit in a form of $U_\pi(s) = r_{s,\pi(s)} + \beta \sum_{s'} p^{\prime(s)}_{s,s'} U_\pi(s')$, where $r_{s,\pi(s)}$ means the reward the process will get in the state s, $\pi(s)$ means taking a certain action and s' is a state that state s can transit to.

Based on the two formulas, we can use **Bellman's Equation** to obtain the optimal utility for a certain state, which is defined as

$U^*(t) = \max_{a \in A(s)}[r_{s,a} + \beta \sum_{s'} p^a_{s,s'} U^*(s')]$, where $A(s)$ is a finite set of all possible actions.

Similarly, we can obtain the optimal policy in form of

$\pi^*(s) = \arg\max_{a \in A(s)}[r_{s,a} + \beta \sum_{s'} p^a_{s,s'} U^*(s')]$.

Then, based on those formulas above, we can apply technique of value iteration. And as the value iteration process reaches a point where $\sum_s U_t(s) - \sum_s U_{t-1}(s) < \epsilon$, which means all the utilities converge to a certain value, we can obtain those utilities as the optimal utility, and so is to obtain the optimal policies. In the program, I put the value of ϵ as 10^{-10}.

1.1 Question a)

Solution:

According to the results I obtained by running the program, the optimal utility for each state is:

The optimal utility for the **New state** is **800.5316098760615**.

The optimal utility for the states from **Used1 to Used8** are **778.3684554178556, 643.2222947711231, 556.1235696440165, 502.836002845536, 475.84600363598634, 470.4784488884467, 470.4784488884467, 470.4784488884467**.

The optimal utility for the **Dead state** is **470.4784488884467**.

```
The optimal utility for the New state is: 800.5316098760615
And the optimal policy is: GO ON USING
The optimal utilities for the states from Used1 to Used8 are: [778.3684554178556, 643.2222947711231, 556.1235696440165,
502.836002845536, 475.84600363598634, 470.4784488884467, 470.4784488884467, 470.4784488884467]
And the optimal policies are: ['GO ON USING', 'GO ON USING', 'GO ON USING', 'GO ON USING', 'GO ON USING', 'GET A NEW ONE
', 'GET A NEW ONE', 'GET A NEW ONE']
The optimal utility for the state Dead is: 470.4784488884467
And the optimal policy is: GET A NEW ONE
```

1.2 Question b)

Solution:

Based on the results I obtained by running the program, the optimal policy for each state is:

The optimal policy for the New state is **to go on using the machine**.

The optimal policies for the states from Used1 to Used5 are **to go on using the machine**, and the optimal policies for the states from Used6 to Used8 are **to replace a new machine**.

The optimal policy for the state Dead is **to replace a new machine**.

```
The optimal utility for the New state is: 800.5316098760615
And the optimal policy is: GO ON USING
The optimal utilities for the states from Used1 to Used8 are: [778.3684554178556, 643.2222947711231, 556.1235696440165,
502.836002845536, 475.84600363598634, 470.4784488884467, 470.4784488884467, 470.4784488884467]
And the optimal policies are: ['GO ON USING', 'GO ON USING', 'GO ON USING', 'GO ON USING', 'GO ON USING', 'GET A NEW ONE
', 'GET A NEW ONE', 'GET A NEW ONE']
The optimal utility for the state Dead is: 470.4784488884467
And the optimal policy is: GET A NEW ONE
```

1.3 Question c)

Solution:

In order to answer this question, I modified my program and add two updated functions to calculate bellman equation during the whole value iteration process.

Specifically, based on the given information, I introduce the utility of the offer provided by the MachineSellingBot as : $U(offer) = \beta * (0.5 * U_\pi(Used1) + 0.5 * U_\pi(Used2)) - cost$, where π is a corresponding policy, to reflect the changes right here, where *cost* is the highest price that a buyer would accept to buy a used machine. In order to find this cost, based on the known two extreme cases (if the cost is as much as that of buying a new machine, then the buyer would definitely buy a new machine rather than a used one; if the cost becomes 0, buying a used machine would be definitely prioritized over buying a new machine), I make the cost iterates from 250 to 0 to find out the highest price that the buyer would choose a used machine. Based on this cost, the utility of buying a used machine can be calculated. Now for each step in value iteration process, instead of comparing the utility of buying a new machine with that of going on using current machine, the utility of buying a used machine is introduced for comparison. As soon as the action of buying a used machine is in the optimal policy, the cost is the highest price that one regards buying a used machine as the rational choice.

After running the program, I got this cost as much as 169.

```
The highest price that buying a used machine would still be the rational choice is:169
```

1.4 Question d)

Solution:

In order to find the optimal policy for all sufficiently large β, I make a series experiments on different β values range from $0.1, 0.3, 0.5$, to $1-0.3^i$, i varies from 1 to 8. All the experiment data I collected from the program is demonstrated as below.

Results		
Beta/Discount	Optimal Policy(Use U to represent 'GO ON USING' and 'R' to represent 'replace')	Optimal Utilities(To demonstrate better, all numbers are rounded to integers)
0.1	[U, U, U, U, U, U, U, U, U, R]	[110, 100, 89, 77, 66, 55, 44, 31, 1, -239]
0.3	[U, U, U, U, U, U, U, U, U, R]	[138, 128, 113, 98, 83, 68, 51, 26, -32, -208]
0.5	[U, U, U, U, U, U, U, U, U, R]	[189, 178, 156, 133, 110, 85, 56, 16, -47, -156]
1 - 0.3	[U, U, U, U, U, U, U, U, U, R]	[303, 290, 247, 203, 161, 118, 22, 37, -1, -38]
$1 - 0.3^2$	[U, U, U, U, U, U, R, R, R, R]	[878, 855, 712, 624, 573, 550, 549, 549, 549, 549]
$1 - 0.3^3$	[U, U, U, U, R, R, R, R, R, R]	[2654, 2625, 2429, 2355, 2333, 2333, 2333, 2333, 2333, 2333]
$1 - 0.3^4$	[U, U, U, U, R, R, R, R, R, R]	[8540, 8509, 8297, 8232, 8221, 8221, 8221, 8221, 8221, 8221]
$1 - 0.3^5$	[U, U, U, U, R, R, R, R, R, R]	[28140, 28109, 27891, 27830, 27822, 27822, 27822, 27822, 27822, 27822]
$1 - 0.3^6$	[U, U, U, U, R, R, R, R, R, R]	[93467, 93436, 93217, 93156, 93149, 93149, 93149, 93149, 93149, 93149]
$1 - 0.3^7$	[U, U, U, U, R, R, R, R, R, R]	[311222, 311190, 310971, 310911, 310904, 310904, 310904, 310904, 310904, 310904]
$1 - 0.3^8$	[U, U, U, U, R, R, R, R, R, R]	[1037071, 1037039, 1036820, 1036760, 1036753, 1036753, 1036753, 1036753, 1036753, 1036753]

1.5 Bonus

Solution:

As shown in the question a), with the cost of a new machine being 250, the optimal utility of state New, or surely using a new machine, is around 800.53, which is also the long term discounted value of a new machine as implied in question a).

To make a person operates a new machine at a net loss, in other words the person is not satisfied of using the new machine, then the price should be higher than 800.53. On the other side, if one operates a new machine at a net gain, then the utility of using the new machine should be smaller than 800.53.