Homework 2
Dynamic Programming

Surabhi S Nath
2016271

Ques 2

Here, we need to solve for V values using linear equations.

V(s) is given by:

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big], \quad \text{for all } s \in \mathcal{S},$$

For our case, p(s',r|s,a) is 1 since every action takes you to a unique state.

Now based on the 4 possible actions - Up, Down, Right, Left, the new states s' will be 4 different states in case of middle cells, 3 different cells and one s' = s in case of edge cells and 2 different cells and two s' = s in case of corner cells.

We can thus write a general expression for V(s) in terms of V(s') in the form of
V(s) = a V(s'1) + b V(s'2) + c V(s'3) + d V(s'4) + b

The coefficients a, b, c, d can be represented through a coefficient matrix A. The V(si) values are the x matrix and b values or constant terms are the bias value which make up the vector b.

Thus we need to solve:
Ax = b to find our x or V(s) values

General expression for corner cell
V(s) = discount * 0.25 V(s'1) + discount * 0.25 V(s'2) + discount * 0.5 - 1 V(s) + 0.5

General expression for edge cells

V(s) = discount * 0.25 V(s'1) + discount * 0.25 V(s'2) + discount * 0.25 V(s'3) + discount * 0.25 - 1 V(s) + 0.25

General expression for middle cells
V(s) = discount * 0.25 V(s'1) + discount * 0.25 V(s'2) + discount * 0.25 V(s'3) + -1 V(s)

Using this, the V(s) obtained exactly matched the values in the book

Ques 4

Here, since we need to find V*(s) values, we need to take max over returns for each of the 4 action. The equation for V*(s) is given by:

$$= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].$$

Again, p(s', r | s, a) is 1.

Here, we convert the nonlinear problem into a linear one by writing it as 4 different equations. After this, we see that V(s) needs to be greater than all other V(s') thus we need to solve the equation Ax >= b here.

Using scipy linprog, we can obtain the optimal V* values. Then optimal action or policy is found out for each state. It is seen that both V* values and policy directions matched the figures in the book.

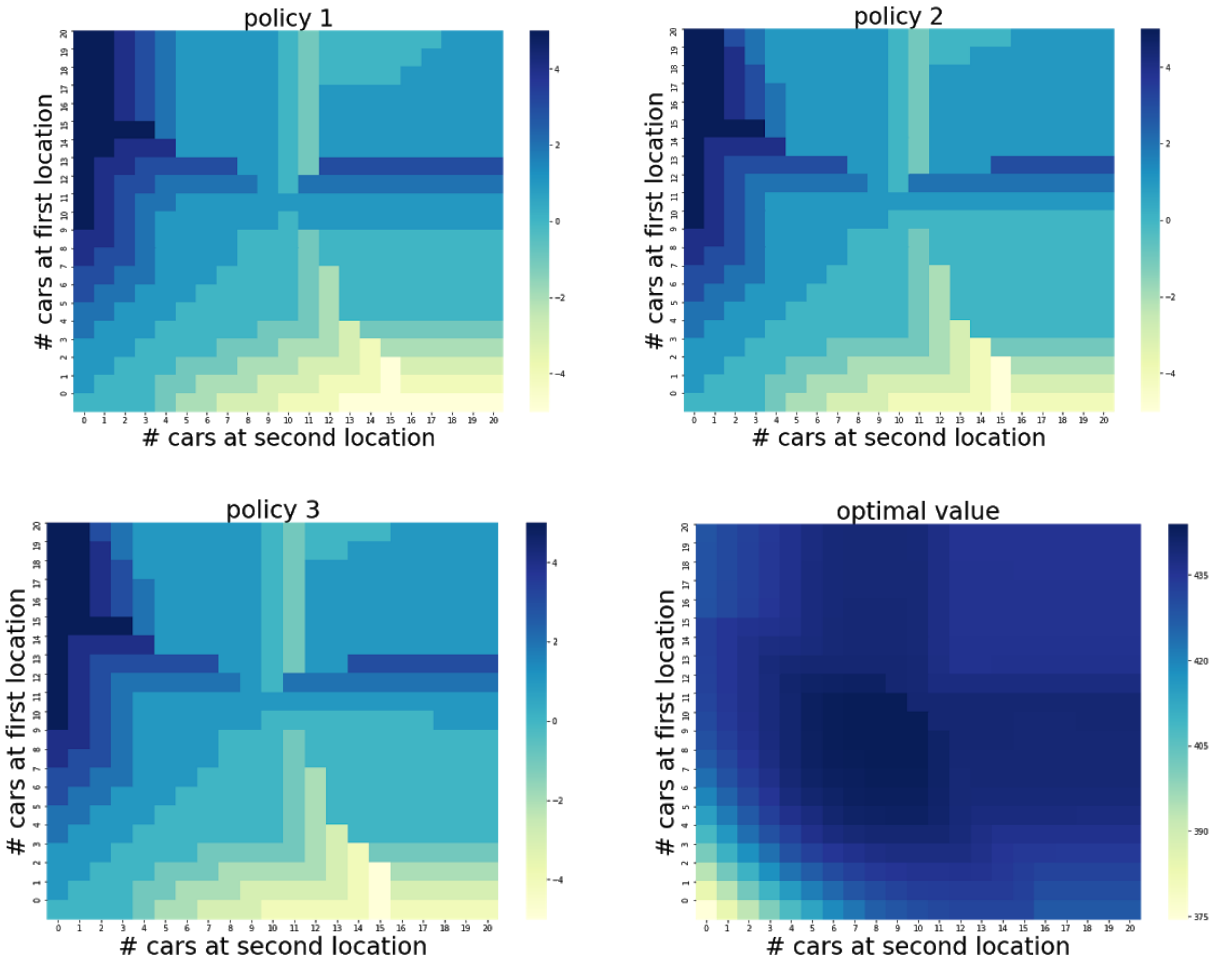
Ques 6

The code performs policy iteration and value iteration. Policy iteration includes policy evaluation and policy improvement. It is seen that the V values and policy from both methods were the same. The bug in the given code can be solved by taking policy to be probabilistic and not fixed to be one particular action. Ie, pi(s|a) instead of pi(s) formulation. This will ensure convergence

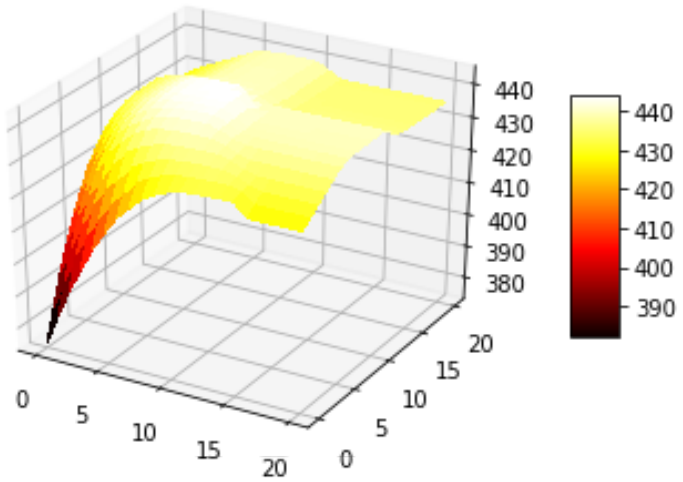
Ques 7

Plots obtained at lim = 10

Policy plots:



3D plot:



The modified car rental problem introduces a free car transport and also limits the number of cars at both locations to 10 if not assigns a penalty. Due to this it is seen that more number of cars are transported as compared to the regular problem. Also, the policy graphs show a higher density in the middle as compared to the normal case.