Projet de Renforcement Learning

Jed Moutahir

December 2023

1 Introduction

The parameters chosen for the following are :
$\lambda = 0.3$, $\mu_1 = 0.2$, $\mu_2 = 0.4$, $\gamma = 0.99$, $Q_1 = 20$, $Q_2 = 20$
The code is available at this Github Folder

2 MDP

2.1 Policy Evaluation

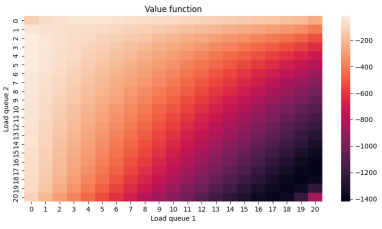
Assume the random policy that dispatches every job with probability 0.5 to either queue 1 or 2.

2.1.1 Bellman's equation

Here is Bellman's equation that characterizes the value function for this policy :

$$V[i,j] = -(i+j)+\gamma*(\frac{\lambda}{2}*(V[i+1,j]+V[i,j+1])+\mu_1*V[i-1,j]+\mu_2*V[i,j-1])+(1-\lambda-\mu_1-\mu_2)*V[i,j]$$

2.1.2 Value function



2.2 Optimal control

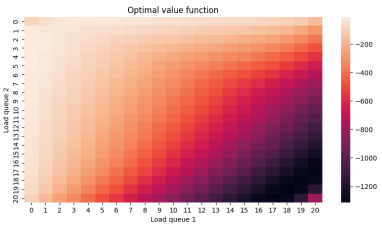
In this part you are asked to find the optimal policy to dispatch incoming jobs.

2.2.1 Bellman's equation

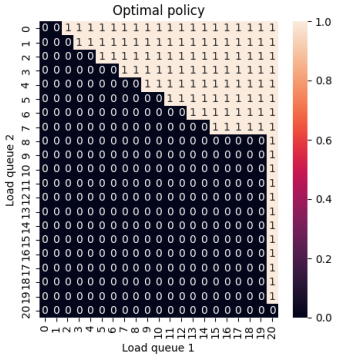
Here is Bellman's equation that characterizes the value function for this policy :

$$V[i,j] = -(i+j)+\gamma*(\lambda*max(V[i+1,j],V[i,j+1])+\mu_1*V[i-1,j]+\mu_2*V[i,j-1])+(1-\lambda-\mu_1-\mu_2)*V[i,j]$$

2.2.2 Value function

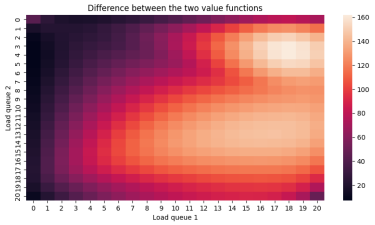


2.2.3 Optimal Policy



2.2.4 Comparing performances

As we can see in the following plot, the optimal policy is always better for every state possible.

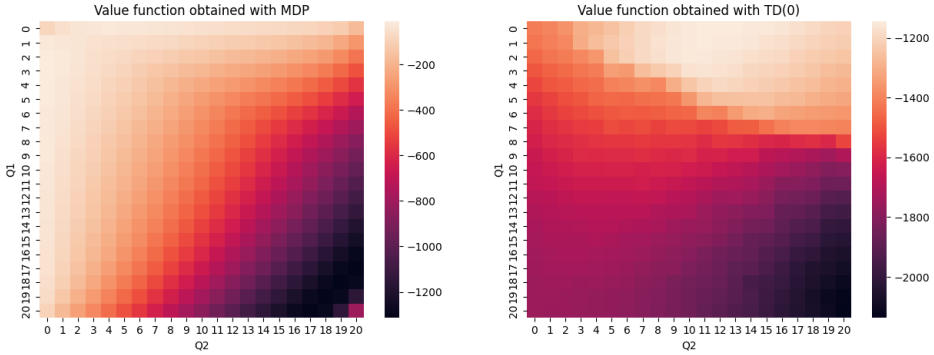


3 Tabular Model-Free control

3.1 Policy Evaluation

3.1.1 Comparing TD(0) and MDP

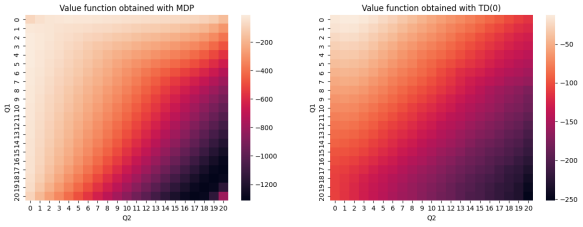
Here are the value functions for TD(0) and MDP :



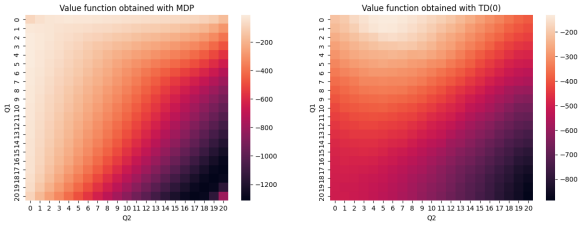
Unfortunately, the value function calculated using TD(0) and $\alpha_n = \frac{1}{n}$ does not correspond really well with the one computed with MDP. Hence, experimenting with α_n could help the convergence of the algorithm.

3.1.2 Experimenting with α_n

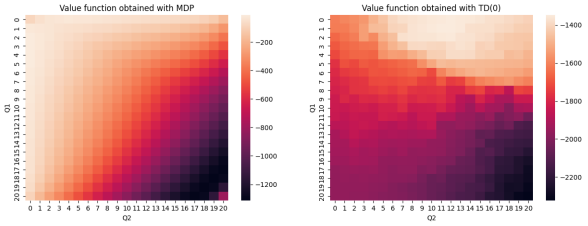
Here are multiple experiments on how $alpha_n$ could help the convergence of TD(0) :
$\alpha = \frac{1}{\sqrt{n}}$



$\alpha = \frac{1}{ln(n)}$



$\alpha = \frac{1}{ln(ln(n))}$



The closest one is found using $\alpha_n = \frac{1}{\sqrt{n}}$

3.2 Optimal control

3.2.1 Q-Learning

Here are the value function and the optimal policy found using Q-Learning and $\alpha_n = 0.1$

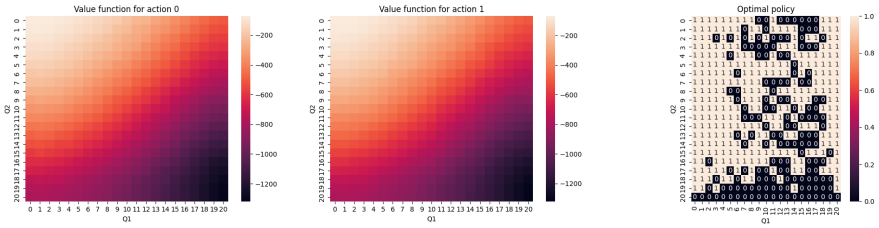
For this problem, Q-Learning seems to be converging at first glance. However, even though the Value function is close to the one computed with MDP, the policy is quite far from the expected one.

3.2.2 Experimenting with α_n

$\alpha_n = \frac{1}{n}$

$\alpha_n = \frac{1}{\sqrt{n}}$

$\alpha_n = \frac{1}{n^\gamma}$ with $\gamma = 0.5$



Even after experimenting with α_n we are not able to make the Q-Learning method converge. This method does not seem to fit this problem.

4 Model-free control with Value Function/Policy approximation

The subject chosen is Policy approximation with softmax parametrization

We now only have 2 parameters :

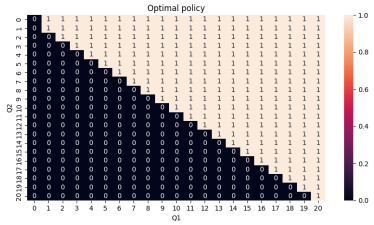
$$\theta_1, \theta_2$$

The policy chosen is :

$$\Pi(a = 1|s = (Q1, Q2)) = \frac{e^{\theta_2 Q1}}{e^{\theta_1 Q2} + e^{\theta_2 Q1}}$$

$$\Pi(a = 0|s = (Q1, Q2)) = \frac{e^{\theta_1 Q2}}{e^{\theta_1 Q2} + e^{\theta_2 Q1}}$$

Here are the results obtained :



Unfortunaltely, because the dimention of the problem was reduced so much, it is impossible to get a great policy. Still, the solution found is quite good.

5 Conclusion

This problem is quite hard to solve using Reinforcement Learning methods other than the Iterative Policy Evaluation.