Markov Decision Problem - Task 4

Gabriel Gattaux

May 2021

1 Introduction

This task was programmed and plot in Matlab. The way to correctly run and compile this task is explained in the *README.txt* file inside the Folder *SolvMarkDecProc* . A file named "appendix.txt" have to be in this folder, to compute the first world , you can change the name of the file but also changed the value of the called function $read_f3$ inside the driver. The termination conditions for the first Part is always $e = 0.0001$. The program on matlab for the First Part was inspired by the java code at *https://galweejit.wordpress.com/2010/12/16/ai-class-implementation-of-mdp-grid-world-from-week-5-unit-9/*.

2 Direct Solving an MDP

2.1 Program verification - 4x3 World

- *Parameters of the World :*

$N = 4; M = 3; p1 = 0.8; p2 = p3 = 0.1; r = -0.04; \gamma = 1; T1 = -1; T2 = 1$
As we can see, the convergence plot is the same than in the lecture, the $policy\_1.txt$ and $Utility\_1.txt$ that we can see in the 1 also.

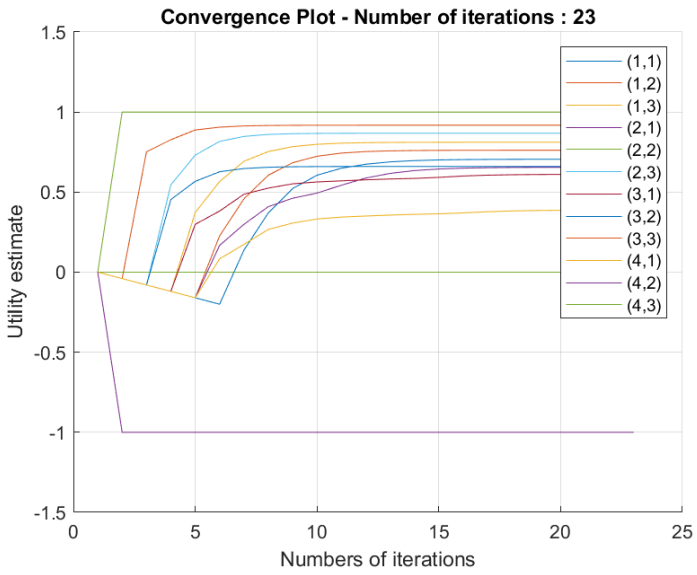


Figure 1: Convergence Plot World 1 4x3

0.8115 >	0.8678 >	0.9178 >	1 +
0.7615 ^	0 #	0.6602 ^	-1 -
0.7052 ^	0.6552 <	0.6113 <	0.3877 <

Table 1: Utility and Policy from World 1

2.2 Problem 2 - 4x4 World

- *Parameters of the World :*

$N = 4; M = 4; p1 = 0.8; p2 = p3 = 0.1; r = -1; \gamma = 0.99; T1 = -20; T2 = 100$

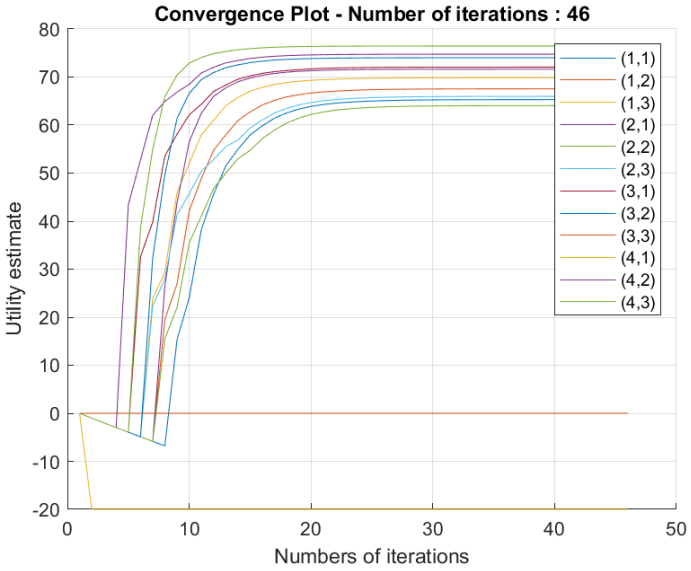


Figure 2: Convergence Plot World 2 4x4 - 1

71.6295 >	74.0185 >	76.4599 >	78.9010 **v**
69.8536 >	72.0652 >	74.7557 ^	81.4650 **v**
67.5415 ^	65.9684 <	-20 -	84.5948 **v**
65.2985 ^	64.0527 ^	0 #	100 +

Table 2: Utility and Policy from World 2

We can observe here that the Value of iteration is greater than the first world due to the size of the problem. And also the policy is made to avoid the Special States, in this case it is almost impossible to go through this case, just in the states (4,2) we have a little probability to go on.

2.3 Problem 3 - 4x4 World

- *Parameters of the World :*

$N = 4; M = 4; p1 = 0.8; p2 = p3 = 0.1; r = -40; \gamma = 0.99; T1 = -20; T2 = 100$

Here the global reward is changed. From R = -1 to R=-40. Our intuition would tell us that it would be better to go in the special state to have a greater reward. And that's what happens. As we can in the Table 3. We can observe that the Number of Iterations is less than before, because the way to go is shorter.

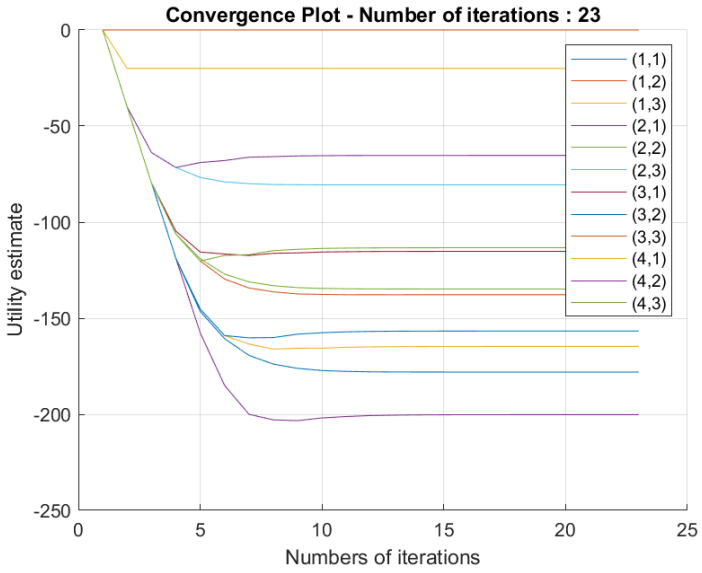


Figure 3: Convergence Plot World 2 4x4 - 3

-200.2063 >	-156.6686 >	-113.3262>	70.2575 **v**
-164.6901 >	-115.1918 >	-65.2820 >	-15.2560 **v**
-137.7545 >	-80.5886 >	-20 **-**	41.3096 **v**
-178.0191 >	-134.7948 ^	0 #	100 +

Table 3: Utility and Policy from World 2 - Pb 3

2.4 Problem 4 - 4x4 World

- *Parameters of the World :*

$N = 4; M = 4; p1 = 0.10; p2 = 0.6; p3 = 0.3; r = -1; \gamma = 0.99; T1 = -20; T2 = 100$

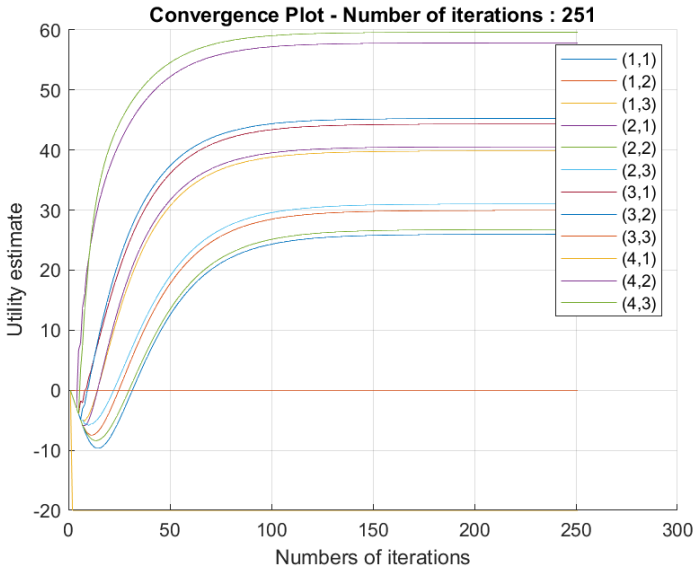


Figure 4: Convergence Plot World 2 4x4 - 4

40.5406 ˆ	45.2728 ˆ	59.6288 >	86.4476 >
39.8558 ˆ	44.3367 ˆ	57.8464 ˆ	89.5865 >
29.9541 ˆ	31.0294 <	-20 -	94.3476 >
25.9744 >	26.7610 >	0 #	100 +

Table 4: Utility and Policy from World 2 - Pb 4

Here the probability to go to the desired motion has changed. So As expected the best way to go to the desired direction is to pass by the right to the chosen action. Like that we are able to never go into the Special States as we can see in the Table 4. However the Number of Iterations is high (251).

2.5 Problem 5 - 4x4 World

- *Parameters of the World :*

$N = 4; M = 4; p1 = 0.8; p2 = p3 = 0.1; r = -1; \gamma = 0.80; T1 = -20; T2 = 100$

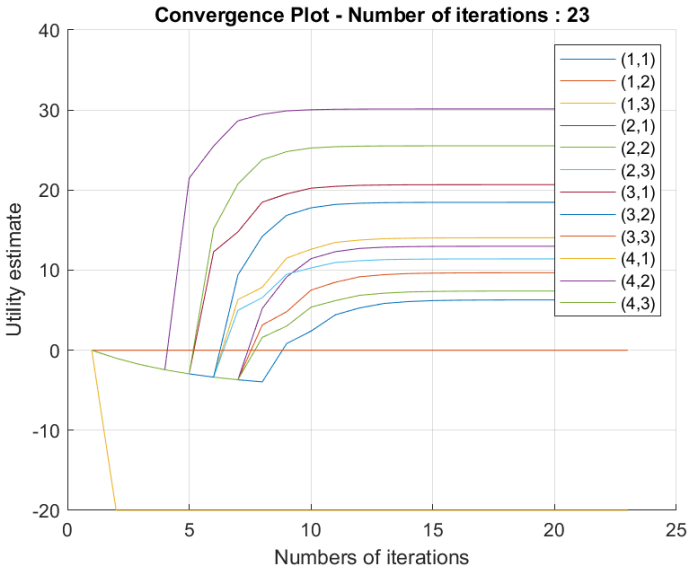


Figure 5: Convergence Plot World 2 4x4 - 4

12.9849 >	18.4724 >	25.5318 >	34.4973 **v**
14.0472 >	20.6785 >	30.1375 >	47.9609 **v**
9.6770 ˆ	11.4084 ˆ	-20 -	66.7391 **v**
6.2879 ˆ	7.3960 ˆ	0 #	100 +

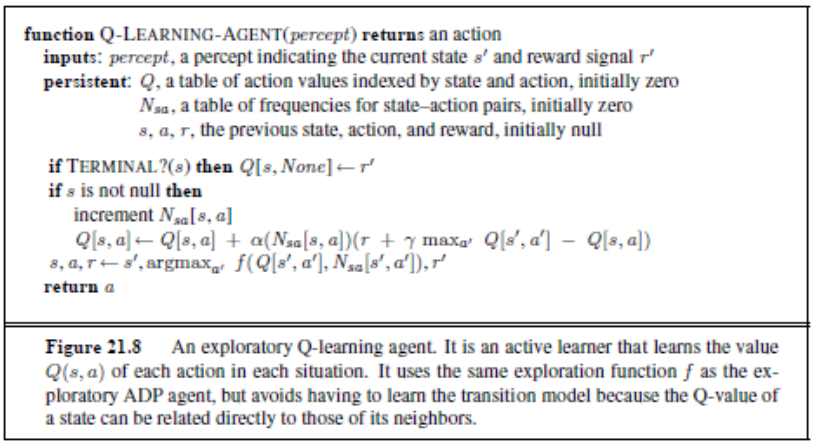
Table 5: Utility and Policy from World 2 - Pb 5

Here γ has changed. Indeed we can see that the Number of iterations has decreased compare to the Figure 2 with the same parameter. We can observe that the policy is more "grouped" and respond to an "basic" algorithm.

3 Q-Learning with Temporal Difference

For this section, I created all the function in Matlab in the path *Part_II_QL*. I began the task with the Temporal Difference function in the Figure 6. And I see after that I had to use the other one,

with Probability. So I will show you my results. The number of iteration for each problem is 100 000. Furthermore, I choose the put the learning parameter $\alpha = 1$, to simply the problem and fit with the specification in the assignment and not changed the algorithm.



function Q-LEARNING-AGENT(*percept*) **returns** an action
inputs: *percept*, a percept indicating the current state s' and reward signal r'
persistent: Q, a table of action values indexed by state and action, initially zero
N_{sa}, a table of frequencies for state–action pairs, initially zero
s, a, r, the previous state, action, and reward, initially null

if TERMINAL?(s) **then** $Q[s, None] \leftarrow r'$
if s is not null **then**
increment $N_{sa}[s, a]$
$Q[s, a] \leftarrow Q[s, a] + \alpha(N_{sa}[s, a])(r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$
$s, a, r \leftarrow s', \text{argmax}_{a'} f(Q[s', a'], N_{sa}[s', a']), r'$
return a

Figure 21.8 An exploratory Q-learning agent. It is an active learner that learns the value $Q(s, a)$ of each action in each situation. It uses the same exploration function f as the exploratory ADP agent, but avoids having to learn the transition model because the Q-value of a state can be related directly to those of its neighbors.

Figure 6: Pseudo-Code Temporal Difference Q-Learning

3.1 Problem 6

Just in order to verifying my program, I tried to compute the policy and utility for the World 1 with $\epsilon = 0.2$. We can see the final Utility and the Policy in the Table 6. We can also see an example of the Q values for the right action in the Table 7. We can also observe that in the Table 7, the Q values are -1 if we want to turn right and go the bad case, and 1 to go to the best case.

0.9005 >	0.9500 >	1 >	0 +
0.8515 ^	0 #	0.9500 ^	0 -
0.8030 ^	0.8515 >	0.9005 ^	0.8515 <

Table 6: Utility and Policy from World 1 - Pb6

0.9	0.9	1	0
0.8	0	-1	0
0.8	0.8	0.8	0.8

Table 7: Q Values final for right action Pb6

3.2 Problem 7

In this problem I choose $\epsilon = 0.02$ and the World 2 (cf Fig.2 assignment). We can see in the Table 8 that the policy is similar to the Table 5. We can observe in the Table 9 that the value for Right Action in (2,2) to go to the bad case is -20, it means that the best action due to Q values are up ant for example, not right. However we can observe something weird because we have Utility value for the F,B and T case.

3.3 Problem 8

For this problem I changed the $\epsilon = 0.5$. Unfortunately, we can observe that the Table 10 and 11 have not changed.

90.1980 >	92.1192 >	94.0598 >	96.0200 **v**
92.1192 >	94.0598 >	96.0200 >	98.0000 **v**
90.1980 ^	92.1192 ^	0.9500 **-**	100.0000 **v**
88.2960 ^	90.1980 ^	0.9005 #	0.8515 +

Table 8: Utility and Policy from World 2 - Pb7

90.2	92.1	94.1	94.1
92.1	94.1	96.0	96.0
90.2	-20.0	0.0	98.0
88.3	88.3	0.0	0.0

Table 9: Q values for right action - Pb7

90.1981 >	92.1192 >	94.0598 >	96.02 **v**
92.1192 >	94.0598 >	96.02 >	98 **v**
90.1980 ^	92.1192 ^	0.9500 **-**	100 **v**
88.2960 ^	90.1980 ^	0.9005 #	0.8515 +

Table 10: Utility and Policy from World 2 - Pb8

90.2	92.1	94.1	94.1
92.1	94.1	96.0	96.0
90.2	-20.0	0.0	98.0
88.3	88.3	0.0	0.0

Table 11: Q values for right action - Pb8

4 Conclusion

This problem was very interesting, particularly with the off-policy model, for me it is similar to the "Swarm Robotics" for example the Ants, Instead of behave with one guys which has to "respawn" at each Episode. The Ants do the same but with the particularity of releasing pheromones for the next ants, and after few steps, all the ants take the best way. In my mind, we did the same thing but with improbable reasoning in the real life. Can we put the Q-Learning Algorithm inside a "swarm" of ants and each episode the algorithm will take an other ants ? The pheromones could be the intensity of light when on ants-robots put his weight on it, and just one sensor in the ants could see where the light is the most dazzling...