INFO8003-1 **Optimal decision making for complex problems**

PART 1 – MOTIVATION

FEBRUARY 2020

s152764 HOCKERS Pierre – s150986 MIFTARI Bardhyl

1 General structure of the code

In the following report, the basic chosen policy is to move always right. All the other basic movement (always move left, up,down) and a uniform random policy can be called for section 2 and section 3. Three different classes are used. The agent class contains all the data needed to go from one state to another and information about the current state. The grid class contains the environment in which the agent is. The game class is the manager code for both classes

2 Section 2

The trajectory of the always go right policy is given by :

```
(((3.0, 0.0), 'RIGHT'), -9)
(((3.0, 1.0), 'RIGHT'), 3.96)
(((3.0, 2.0), 'RIGHT'), 18.6219)
(((3.0, 3.0), 'RIGHT'), -4.851495)
(((3.0, 4.0), 'RIGHT'), -4.8029800499999995)
(((3.0, 4.0), 'RIGHT'), -4.754950249499999)
(((3.0, 4.0), 'RIGHT'), -4.707400747004999)
(((3.0, 4.0), 'RIGHT'), -4.660326739534949)
(((3.0, 4.0), 'RIGHT'), -4.6137234721396)
(((3.0, 4.0), 'RIGHT'), -4.567586237418204)
```

(a) Deterministic trajectory

```
(((3.0, 0.0), 'RIGHT'), -3)
(((0.0, 0.0), 'RIGHT'), 0.99)
(((0.0, 1.0), 'RIGHT'), -2.9402999999999997)
(((0.0, 0.0), 'RIGHT'), -2.910897)
(((0.0, 0.0), 'RIGHT'), 0.96059601)
(((0.0, 1.0), 'RIGHT'), -2.8529701496999995)
(((0.0, 0.0), 'RIGHT'), -2.8244404482029997)
(((0.0, 0.0), 'RIGHT'), -2.7961960437209696)
(((0.0, 0.0), 'RIGHT'), 0.92274469442792)
(((0.0, 1.0), 'RIGHT'), -4.567586237418204)
```

(b) Stochastic trajectory

Figure 1 – Evolution of the trajectory for 10 steps starting at position (3,0) in deterministic and stochastic case ($\beta = 0.5$). Trajectory contains the position before the move, the chosen move and the reward it got after the move

It can be seen that as expected in the deterministic case, the agent moves to the right and stays in the position (3,4) as he can not go further right. Whereas in the stochastic case, a probability of 0.5 sends the agent to position (0,0) when it wanted to go right.

3 Section 3

Similarly to section 2, the always right. In this section, the agents reward is simulated 10 times for 1000 movement and the mean of this result is plotted for each starting position. The value 1000 for the number of timestep to have a good estimation was chosen arbitrarily in the beginning but , as can be seen in Figure 2 and 3, N = 500 could also have been a good choice. Before 500, there are a lot of oscillations still and after that the rewards do not evolve anymore.

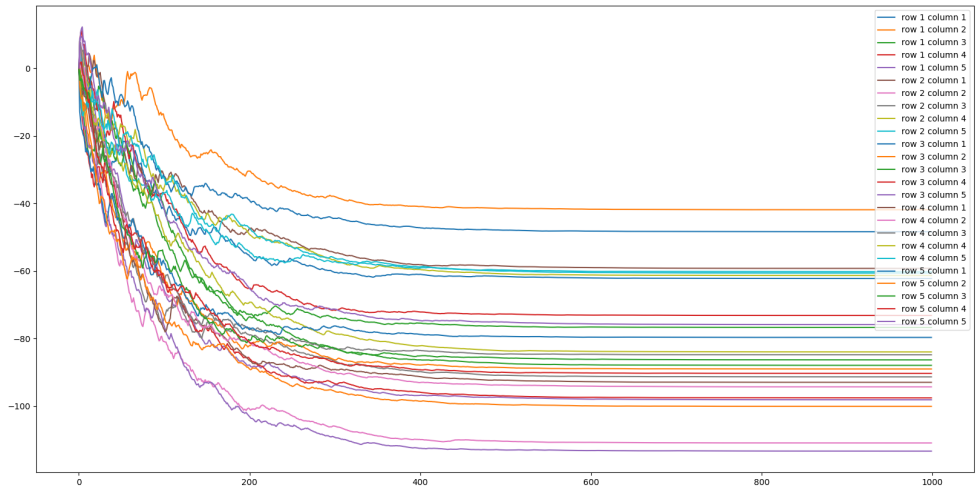

Figure 2 – Deterministic evolution of the rewards over 1000 timesteps for each starting position

In Figure 2, it can be seen that for a deterministic always right policy the column at which the agent starts has a little influence on the expected final reward received as the agent will stay in column 4 once it is there and the rewards to get from the initial state to the column 4 are also different. Whereas, the row has a huge influence on the expected final reward as we can clearly see. If another policy such as always up was used the column would have a huge influence on the final result and the influence of the row would be lesser.

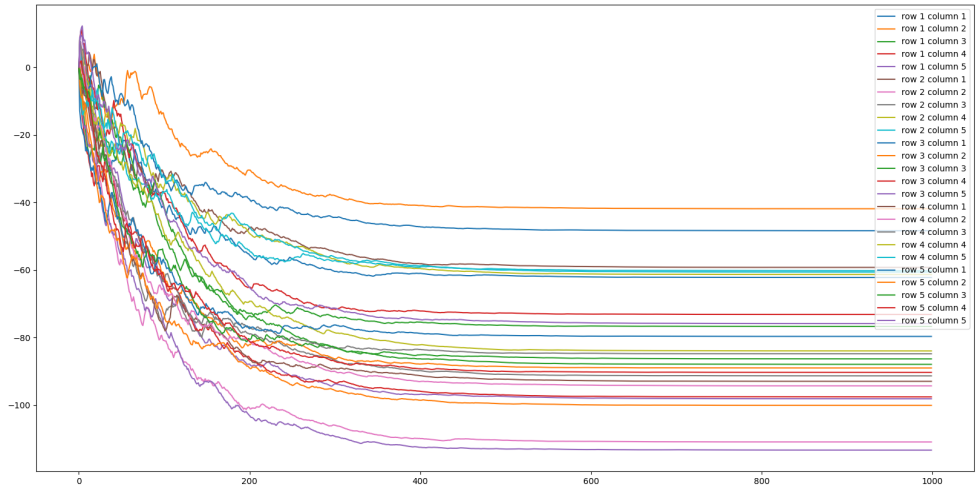

Figure 3 – Stochastic evolution of the rewards over 1000 timesteps for each starting position

In the stochastic case, the starting position does not influence much the expected final reward. For each action that is taken, we have a probability of $\beta = 0.5$ to end up in state (0,0). As we have, $\sum_{n=1}^{\inf}(\frac{1}{2})^n = 1$ the agent will always end up in the (0,0) position which explain why in this case the initial starting position is not that important.

4 Section 4