University of Liège
School of Engineering

Reinforcement Learning in a Discrete Domain

INFO8003-1: Optimal decision making for complex problems

Julien Gustin, Joachim Houyon

February 22, 2022

1 Implementation of the domain

Our rule based policy is to always go right. More formally it can be described as:

$$\mu(x, y) = (1, 0) \;\; \forall x, y \in X \tag{1}$$

Note that in the following for clearness we will represent actions

$$U = \{(1, 0), (-1, 0), (0, 1), (0, -1)\}$$

as follow:

$$U = \{RIGHT, LEFT, UP, DOWN\}$$

Also the state (x, y) does not mean "element at the line x and column y", but element at position [x][y] in the table where (0, 0) is on the top left

-3	1	-5	0	19
6	3	8	9	10
5	-8	4	1	-8
6	-9	4	19	-5
-20	-17	-4	-3	9

Figure 1. Domain instance

Therefore by taking the example of the statement the initial state is at position $(0, 3)$ and not $(3, 0)$.

Other components of the domain remain the same.

1.1 Deterministic

In the deterministic domain the action (1) is always applied to the environment.

Here is a simulation through a single trajectory of 10 steps, starting by the initial state $x_0 = (0,3)$ represented as tuple $(x_0, u_0, r_0, x_1), ..., (x_9, u_9, r_9, x_{10})$

```
--- Deterministic ---

0.  ((0,3), RIGHT, -9, (1,3))
1.  ((1,3), RIGHT, 4, (2,3))
2.  ((2,3), RIGHT, 19, (3,3))
3.  ((3,3), RIGHT, -5, (4,3))
4.  ((4,3), RIGHT, -5, (4,3))
5.  ((4,3), RIGHT, -5, (4,3))
6.  ((4,3), RIGHT, -5, (4,3))
7.  ((4,3), RIGHT, -5, (4,3))
8.  ((4,3), RIGHT, -5, (4,3))
9.  ((4,3), RIGHT, -5, (4,3))
```

Listing 1. Simulated trajectories by applying policy (1) in the deterministic domain.

1.2 Stochastic

In the stochastic domain the action (1) may not be applied due to the noise $w \sim \mathcal{U}(0,1)$. Such that if $w \geq 0.5$ the state is updated to $(x, y) = (0, 0)$.

Here is a simulation through a single trajectory of 10 steps, starting by the initial state $x_0 = (0,3)$ represented as tuple $(x_0, u_0, r_0, x_1), ..., (x_9, u_9, r_9, x_{10})$

```
--- Stochastic ---

0.  ((0,3), RIGHT, -9, (1,3))
1.  ((1,3), RIGHT, -3, (0,0))
2.  ((0,0), RIGHT, -3, (0,0))
3.  ((0,0), RIGHT, -3, (0,0))
4.  ((0,0), RIGHT, 1, (1,0))
5.  ((1,0), RIGHT, -5, (2,0))
6.  ((2,0), RIGHT, 0, (3,0))
7.  ((3,0), RIGHT, -3, (0,0))
8.  ((0,0), RIGHT, -3, (0,0))
9.  ((0,0), RIGHT, -3, (0,0))
```

Listing 2. Simulated trajectories by applying policy (1) in the stochastic domain.

2 Expected return of a policy

As seen on the course there exist a bound of the difference between J_N^μ and J^μ such that

$$||J_N^\mu - J^\mu||_\infty \leq \frac{\gamma^N}{1-\gamma} Br$$

where

$$J_N^\mu(x) = \underset{w \sim P_w(\cdot|x,u)}{E} \left[r(x, \mu(x), w) + \gamma J_{N-1}^\mu(f(x, \mu(x), w)) \right], \quad \forall N \geq 1 \tag{2}$$

with $J_0^\mu(x) = 0$

Therefore we can find a lower bound of N by fixing the error of approximation ϵ, where $\epsilon = ||J_N^\mu - J^\mu||_\infty$.

$$\begin{aligned} \epsilon &\leq \frac{\gamma^N}{1-\gamma} Br \\ \frac{\epsilon(1-\gamma)}{Br} &\leq \gamma^N \\ \log_\gamma\left(\frac{\epsilon(1-\gamma)}{Br}\right) &\leq N \end{aligned}$$

Given $\gamma = 0.99$, $Br = 19$ and $\epsilon = 10^{-6}$ we have:

$$N = \left\lceil \log_\gamma\left(\frac{\epsilon(1-\gamma)}{B_r}\right) \right\rceil = 2126. \tag{3}$$

2.1 Deterministic

$y \backslash x$	**0**	**1**	**2**	**3**	**4**
0	1839.618	1857.190	1881.000	1900.000	1900.000
1	990.040	997.010	999.000	1000.000	1000.000
2	−779.299	−779.090	−791.000	−800.000	−800.000
3	−471.568	−467.240	−476.000	−500.000	−500.000
4	849.369	875.120	888.000	900.000	900.000

Table 1. $J_N^\mu(x, y)$ for all $(x, y) \in X$ in the deterministic domain; $N = 2126$

2.2 Stochastic

$y \backslash x$	**0**	**1**	**2**	**3**	**4**
0	-72.021	-71.456	-64.253	-54.753	-54.753
1	-67.781	-64.911	-64.164	-63.664	-63.664
2	-77.413	-73.258	-76.986	-81.486	-81.486
3	-75.348	-68.075	-66.515	-78.515	-78.515
4	-82.342	-74.124	-70.654	-64.654	-64.654

Table 2. $J_N^{\mu}(x, y)$ for all $(x, y) \in X$ in the stochastic domain; $N = 2126$

3 Optimal policy

We define

$$r(x, u) = \underset{w \sim P_w(\cdot|x,u)}{E}[r(x, u, w)] \quad \forall x \in X, u \in U$$

$$p(x' \mid x, u) = \underset{w \sim P_w(\cdot|x,u)}{E}[I_{\{x'=f(x,u,w)\}}] \quad \forall x, x' \in X, u \in U$$

which defines the structure of a MDP. Thanks to this, we can rewrite the functions Q_N as follows:

$$Q_0(x, u) = 0$$

$$Q_N(x, u) = r(x, u) + \gamma \sum_{x' \in X} p(x' \mid x, u) \underset{u' \in U}{max} Q_{N-1}(x', u') \forall N \geq 1 \tag{4}$$

3.1 Choice of N

We wish to find the smallest N such that

$$||Q_N - Q||_\infty = 0$$

because once we reach this equality, we know that the optimal policy μ^* will be the same as the policy μ_N^* derived from Q_N, ensuring that any increase of N will not modify the policy found.

However, it is not possible to determine such value of N which respects this property, we can only bound the suboptimality of μ_N^* with respect to μ^*:

$$||J^{\mu_N^*} - J^{\mu^*}||_\infty \leq \frac{2\gamma^N B_r}{(1-\gamma)^2}$$

It follows that

$$\underset{N \to +\infty}{lim} \frac{2\gamma^N B_r}{(1-\gamma)^2} = 0$$

and we can only compute a value N such that the gap from optimality is bounded by a value ϵ.

Using $\epsilon = 10^{-6}$, we get that

$$N = \left\lceil \log_{\gamma} \left(\frac{\epsilon(1-\gamma)^2}{2B_r} \right) \right\rceil = 2653. \tag{5}$$

Which is very likely to have $\mu_N^* = \mu^*$ because the rewards are quite big numbers compared to the bound ϵ.

With $J^{\mu_N^*}(x) = \underset{u \in U}{max}\ Q_N(x,u)$ and $\mu_N^* \in arg\ max_{u \in U}\ Q_N(x,u)$

3.2 Deterministic domain

In the deterministic domain, it is clear that

$p(x' \mid x, u) = 1\ \forall x, x' \in X, u \in U$ and $r(x,u) = R(g, F(x,u))\ \forall x \in X, u \in U$

y \ x	**0**	**1**	**2**	**3**	**4**
0	1842.031	1857.190	1881.000	1900.000	1900.000
1	1854.576	1870.279	1881.090	1891.000	1900.000
2	1842.031	1855.576	1870.279	1881.090	1891.000
3	1828.610	1849.010	1863.646	1863.279	1864.090
4	1816.324	1826.520	1849.010	1863.646	1842.010

Table 3. $J_{\mu^*}^N(x,y)$ for all $(x,y) \in X$ in the deterministic domain; $N = 2653$

y \ x	**0**	**1**	**2**	**3**	**4**
0	DOWN	RIGHT	RIGHT	RIGHT	RIGHT
1	RIGHT	RIGHT	RIGHT	RIGHT	UP
2	UP	RIGHT	UP	UP	UP
3	UP	RIGHT	RIGHT	UP	UP
4	UP	RIGHT	UP	UP	LEFT

Table 4. $\mu_N^*(x,y)$ for all $(x,y) \in X$, in the deterministic domain; $N = 2653$

3.3 Stochastic domain

In the stochastic domain, we have the following:

$$r(x,u) = wR(g, F(x,u)) + (1-w)R(g,(0,0)) \; \forall x \in X, u \in U$$

$$p(x' \mid x,u) = w(I_{\{x'=F(x,u)\}}) + (1-w)I_{\{x'=(0,0)\}} \;\; \forall x, x' \in X, u \in U$$

y \ x	**0**	**1**	**2**	**3**	**4**
0	159.446	159.637	163.052	172.130	172.130
1	159.637	163.052	164.903	167.630	172.130
2	159.446	160.137	163.052	167.213	167.630
3	159.259	162.196	167.213	162.196	167.213
4	159.259	155.713	162.196	167.213	162.229

Table 5. $J^N_{\mu^*}(x,y)$ for all $(x,y) \in X$ in the stochastic domain; $N = 2653$

y \ x	**0**	**1**	**2**	**3**	**4**
0	DOWN	DOWN	DOWN	RIGHT	RIGHT
1	RIGHT	RIGHT	RIGHT	RIGHT	UP
2	UP	RIGHT	UP	DOWN	UP
3	LEFT	RIGHT	RIGHT	LEFT	LEFT
4	UP	RIGHT	UP	UP	RIGHT

Table 6. $\mu^*_N(x,y)$ for all $(x,y) \in X$, in the stochastic domain; $N = 2653$

4 System Identification

In order to estimate $r(x,u)$ and $p(x'|x,u)$ from a given trajectory $h_t = (x_0, u_0, r_0, x_1, u_1, r_1, ..., u_{t-1}, r_{t-1}, x_t)$ one can compute it by

$$\hat{r}(x,u) = \frac{1}{|A_h(x,u)|} \sum_{i \in A_h(x,u)} r_i, \tag{6}$$

$$\hat{p}(x' \mid x,u) = \frac{1}{|A_h(x,u)|} \sum_{i \in A_h(x,u)} I_{\{x_{i+1}=x'\}}, \tag{7}$$

where $A_h(x,u)$ is the set of indices $\{i \mid x_i = x, u_i = u\}$. However, the number of operations to estimate the MDP structure grows linearly with t. And also the memory requirement we have therefore decided to implement the algorithm described at the slide 29 of the course.

At time 0, set $N(x,u) = 0$, $N(x,u,x') = 0$, $R(x,u) = 0$, $p(x'|x,u) = 0$, $\forall x, x' \in X$ and $u \in U$.
At time $t \neq 0$, do
1. $N(x_{t-1}, u_{t-1}) \leftarrow N(x_{t-1}, u_{t-1}) + 1$
2. $N(x_{t-1}, u_{t-1}, x_t) \leftarrow N(x_{t-1}, u_{t-1}, x_t) + 1$
3. $R(x_{t-1}, u_{t-1}) \leftarrow R(x_{t-1}, u_{t-1}) + r_t$
4. $r(x_{t-1}, u_{t-1}) \leftarrow \frac{R(x_{t-1}, u_{t-1})}{N(x_{t-1}, u_{t-1})}$
5. $p(x|x_{t-1}, u_{t-1}) \leftarrow \frac{N(x_{t-1}, u_{t-1}, x)}{N(x_t, u_t)} \quad \forall x \in X$

We have then apply this algorithm from a given trajectory h_t starting from $(0,3)$ and then using a random uniform policy. With increasing value of t to show the convergence, $t = \{10^0, 10^1, 10^2, ...10^7\}$.

Note that in the following

$$J^{\hat{\mu}^*_N}(x) = \max_{u \in U} \hat{Q}_N(x,u)$$

4.1 Deterministic

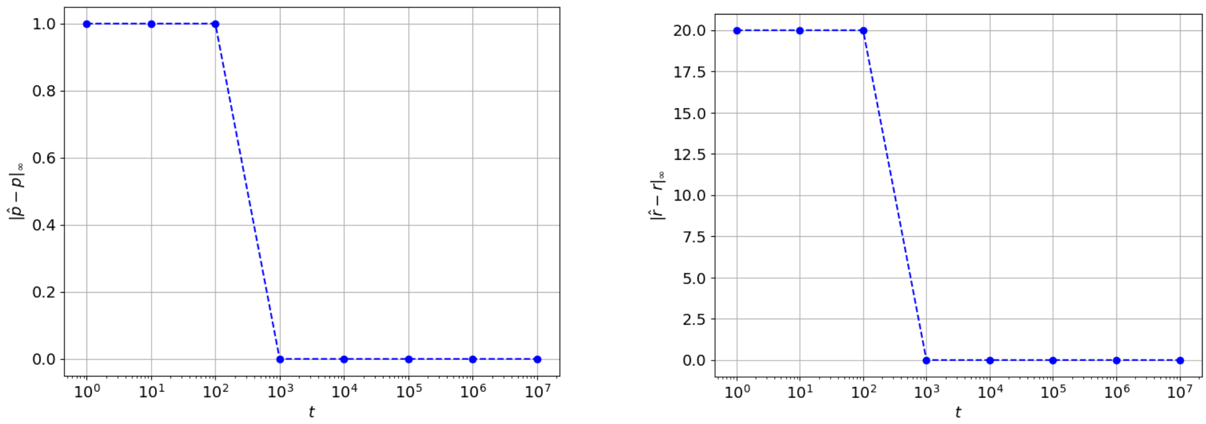

Figure 2. Convergence speed of $\hat{p}$ and $\hat{r}$ towards p and r using the ∞ norm through a growing trajectory generated by a random uniform trajectory in a deterministic domain

Clearly in the deterministic domain $\hat{p}$ and $\hat{r}$ converge after a trajectory length of 10^3. Which are then use to compute the $\hat{Q}_N$-functions like the Q_N (4) one by just replacing r by $\hat{r}$ and p by $\hat{p}$. Therefore if $\hat{p}$ and $\hat{r}$ has converged $\hat{Q}_N$ will also as we can see just below.

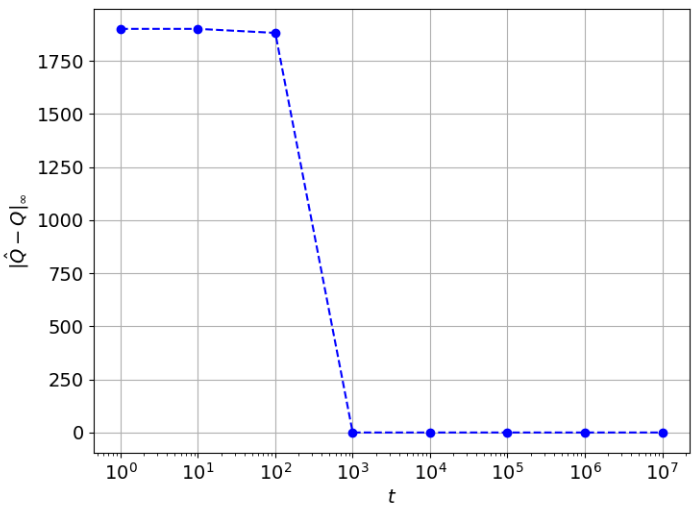

Figure 3. Infinity norm between $\hat{Q}_N$ and Q_N for each trajectory length

$y \backslash x$	**0**	**1**	**2**	**3**	**4**
0	1842.031	1857.190	1881.000	1900.000	1900.000
1	1854.576	1870.279	1881.090	1891.000	1900.000
2	1842.031	1855.576	1870.279	1881.090	1891.000
3	1828.610	1849.010	1863.646	1863.279	1864.090
4	1816.324	1826.520	1849.010	1863.646	1842.010

Table 7. $J^{\mu_N^*}(x, y)$ for all $(x, y) \in X$ in the deterministic domain; $N = 2653$

Using $\hat{Q}_N$ we can then approximate $\hat{\mu}_N^*$ such that $\hat{\mu}_N^* \in arg\ max_{u \in U}\ \hat{Q}_N(x, u)$. And in the following we computed $\hat{Q}_N$ with a trajectory length of 10^7 steps.

$y \backslash x$	**0**	**1**	**2**	**3**	**4**
0	1842.031	1857.190	1881.000	1900.000	1900.000
1	1854.576	1870.279	1881.090	1891.000	1900.000
2	1842.031	1855.576	1870.279	1881.090	1891.000
3	1828.610	1849.010	1863.646	1863.279	1864.090
4	1816.324	1826.520	1849.010	1863.646	1842.010

Table 8. $J^{\hat{\mu}_N^*}(x, y)$ for all $(x, y) \in X$ in the deterministic domain; $N = 2653$

$y \backslash x$	**0**	**1**	**2**	**3**	**4**
0	DOWN	RIGHT	RIGHT	RIGHT	RIGHT
1	RIGHT	RIGHT	RIGHT	RIGHT	UP
2	UP	RIGHT	UP	UP	UP
3	UP	RIGHT	RIGHT	UP	UP
4	UP	RIGHT	UP	UP	LEFT

Table 9. $\hat{\mu}_N^*(x, y)$ for all $(x, y) \in X$ in the deterministic domain; $N = 2653$
.

We can see that as expected the result of Table 7. and Table 8. are indeed the same.

4.2 Stochastic

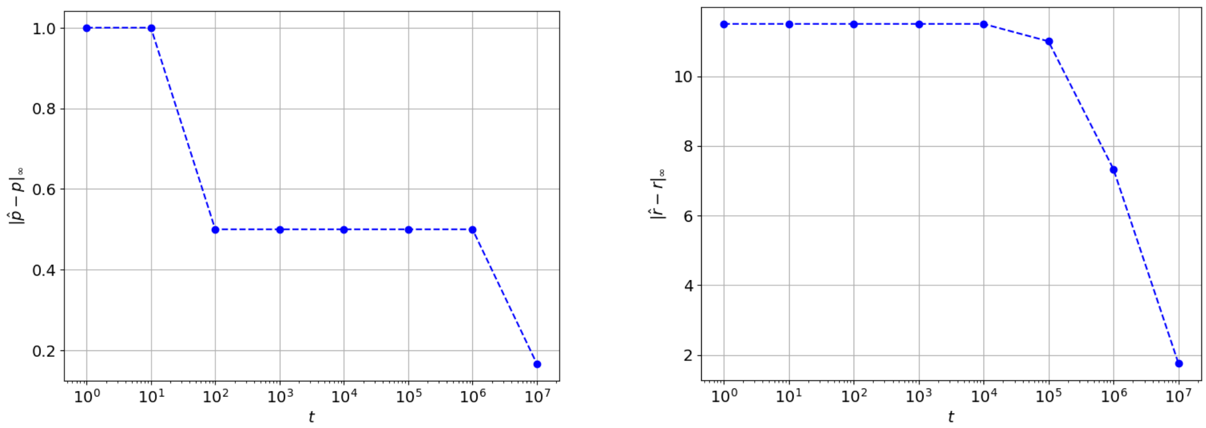

Figure 4. Convergence speed of $\hat{p}$ and $\hat{r}$ towards p and r using the ∞ norm through a growing trajectory generated by a random uniform trajectory in a stochastic domain

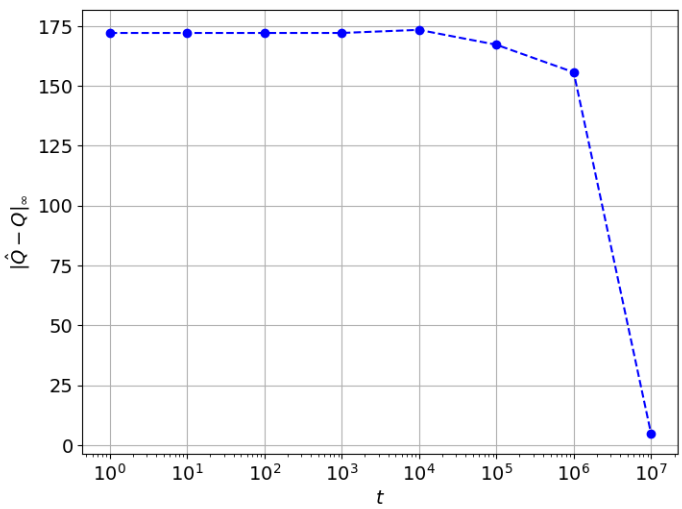

Figure 5. Infinity norm between $\hat{Q}_N$ and Q_N for each trajectory length

Here $\hat{r}$ and $\hat{p}$ are close to the true value but have not converged after a trajectory length of 10^7 so $\hat{Q}_N$ has not converged either. As before in the following we computed $\hat{Q}_N$ with a trajectory length of 10^7 steps.

y \ x	**0**	**1**	**2**	**3**	**4**
0	159.446	159.637	163.052	172.130	172.130
1	159.637	163.052	164.903	167.630	172.130
2	159.446	160.137	163.052	167.213	167.630
3	159.259	162.196	167.213	162.196	167.213
4	159.259	155.713	162.196	167.213	162.229

Table 10. $J^{\mu^*_N}(x, y)$ for all $(x, y) \in X$ in the stochastic domain; $N = 2653$

y \ x	**0**	**1**	**2**	**3**	**4**
0	158.222	158.405	161.846	171.526	171.860
1	158.401	161.815	163.586	165.930	170.465
2	158.214	159.019	162.080	166.025	165.946
3	158.032	161.146	166.322	160.963	166.769
4	157.772	154.509	161.048	166.820	167.076

Table 11. $J^{\hat{\mu}^*_N}(x, y)$ for all $(x, y) \in X$ in the stochastic domain; $N = 2653$

y \ x	**0**	**1**	**2**	**3**	**4**
0	DOWN	DOWN	DOWN	RIGHT	UP
1	RIGHT	RIGHT	RIGHT	RIGHT	UP
2	UP	RIGHT	UP	DOWN	UP
3	LEFT	RIGHT	RIGHT	LEFT	LEFT
4	UP	RIGHT	UP	UP	DOWN

Table 12. $\hat{\mu}^*_N(x, y)$ for all $(x, y) \in X$ in the deterministic domain; $N = 2653$

We can notice that, in the stochastic configuration, the convergence is way slower.

To explain the influence of the length of the trajectory on the quality of the approximations, we can compare the results with the deterministic domain and the stochastic domain.

It is trivial that the longer is the trajectory, the more likely it will contain a transition $(x, u, r) \rightarrow x'$ of the domain.

Furthermore, the strong law of large numbers states that the more samples we have (in our case, a sample is a transition $(x, u, r) \rightarrow x'$ of the domain), the closer we should get from the real mean of $r(.)$ and $p(.)$.

We can also explain the big difference of convergence between the deterministic domain and the stochastic domain:

In the deterministic domain, it is easier to reach any state x' from a state x than in the stochastic domain, because, at each transition in the stochastic configuration, there is a probability of 0.5 to be teleported back to the state $(0, 0)$. Therefore, the probability of reaching states like the state $(4, 4)$ is very low, and so it is unlikely for a small trajectory to contain such states, and by the strong law of large number, moreover these state much be reach a certain amount of time in order to be close to the true value, while for the deterministic domain a single pass is enough to get the true value.