DM M2 ORCO Optimization under uncertainty

Romain Cravic, Bruno Gaujal

October 2023

The following problem is inspired from the board game "Gods love dinosaurs".

1 Problem statement

The goal of this homework is to design an optimal management of the food chain to make dinosaurs prosperate in the world. We suppose there are three types of animals :

- The rabbits (R) can reproduce on their own as long as there are some rabbits left.
- The tigers (T) have to eat rabbits to survive and reproduce.
- The dinosaurs (D) have to eat either rabbits or tigers to survive. They lay eggs only if they eat tigers.

The problem is modeled as follows. There are N cells arranged as a circle as shown in Figure 1. At each time t and in each cell there is either nothing, or a rabbit, or a tiger (but no dinosaur).

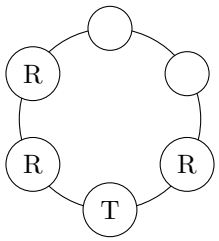

Figure 1 : $N = 6$

You are god and at each time t you choose one of the following actions :

- **Activate Rabbits (AR)** : All the rabbits produce a new rabbit in their two neighbour cells if they are currently empty. See Figure 2.
- **Activate Tigers (AT)** : All the tigers move forward clockwise to eat the rabbits and reproduce in the two next cells. The cells that currently contain a rabbit are replaced by a tiger (so 0, 1, or 2 tigers appear), and the start cell of the tiger becomes empty. See Figure 3.
- **Activate Dinosaur (AD)** : The dinosaur eats the content of $K \geq 1$ random cells picked with a uniform probability. See Figure 4 for $K = 3$.
- **Birth Rabbit (BR)** : If there is at least one empty cell, a rabbit appears in one random empty cell picked with the uniform probability.
- **Birth Tiger (BT)** : If there is at least one empty cell, a tiger appears in one random empty cell picked with the uniform probability.

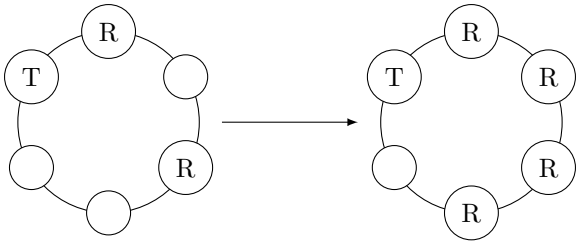

Figure 2 : (AR)

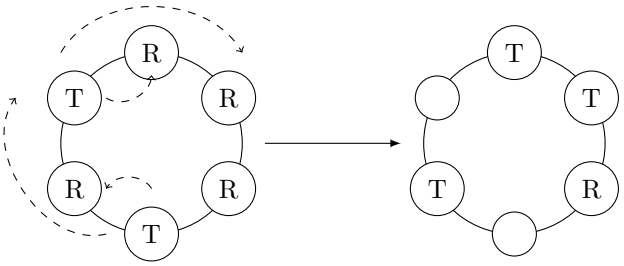

Figure 3 : (AT)

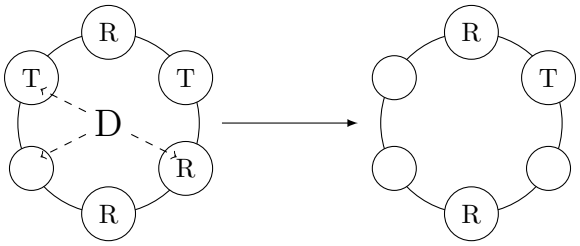

Figure 4 : (AD)

Whenever a dinosaure eats a tiger, it lays an egg and you win W points. However, whenever you activate the dinosaure and it does not eat anything, you endanger the species so you lose L points. Finally, giving birth to a rabbit or a tiger has a cost C_R and C_T respectively.

2 Preliminaries

We want to model the above problem as a MDP $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P)$.

Question 1 : Describe the state space $\mathcal{S}$ and the action space $\mathcal{A}$. What is the size of the state space as a function of N (without eliminating symmetries) ?

Question 2 : For the particular state s illustrated in Figure 1, and for $K = 3$, describe for each action a what are the s' that may be the next state of the MDP with positive probability, ie s' such that $P(s'|s, a) > 0$.

Question 3 Write a pseudo-code for the reward function $\mathcal{R}(s, a, s')$ that returns the immediate reward if the environment is in state s, the action a is taken, and the next state is s'. *Hint : the knowleadge of s' is necessary only for action (AD). In this case you may first check that s' is a plausible next state for (s, AD) (otherwise return 0), and then distinguish the different scenarios (egg, species endangered, ...).*

Note : During the course, you always saw $R(s, a)$ instead of $R(s, a, s')$ for the reward function. In the next section you'll prove this is not a problem to have a dependancy in s'.

3 Theoretical part

3.1 Finite-horizon setting

We first consider the problem has a finite-horizon T. The goal is to maximize $\sum_{t=1}^{T} R(s_t, a_t, s_{t+1})$.

For any Markovian policy $\pi = \{\pi_t : \mathcal{S} \to \mathcal{A}\}_{t \in 1 \dots T}$ and $\tau \in \{1, \dots, T\}$, we define the value function of π from time τ as

$$\forall s, \quad V_\tau^\pi(s) = \mathbb{E}\left[\sum_{t=\tau}^{T} R(s_t, \pi_t(s_t), s_{t+1}) \middle| s_\tau = s \right]$$

and $\forall s, \quad V_{T+1}^\pi(s) = 0$.

Question 4 : Show that for every $\tau \leq T$ and every state s, the following equality holds :

$$V_\tau^\pi(s) = \sum_{s'} P(s'|s, \pi_\tau(s)) \left[R(s, \pi_\tau(s), s') + V_{\tau+1}^\pi(s') \right]$$

We denote by V_τ^* the optimal value functions, ie $\forall s, \tau, \quad V_\tau^*(s) = \max_\pi V_\tau^\pi(s)$.

Question 5 : Show that V^* satisfies the following Bellman's equation :

$$\forall s, \tau \quad V_\tau^*(s) = \max_a \sum_{s'} P(s'|s, a) \left[R(s, a, s') + V_{\tau+1}^*(s') \right]$$

Question 6 : Explain how we can compute V_τ^* for every τ, and an optimal policy $\pi^* = \{\pi_\tau^*\}$ that satisfies $\forall s, \tau, \quad V_\tau^{\pi^*}(s) = V_\tau^*(s)$.

3.2 Infinite-horizon average-reward setting

In the following we consider the average-reward objective for our problem, that is we want to maximize the average-reward g :

$$g = \lim_{T \to +\infty} \frac{1}{T} \sum_{t=1}^{T} \mathcal{R}(s_t, a_t, s_{t+1})$$

We first need to study the structure of the MDP.

Question 7 : Show that the MDP is communicating. *Hint : Describe an action sequence that enables to get a positive probability path to go from an arbitrary state s to an other arbitrary state s'.*

Question 8 : Show that the MDP is not unichain. *Hint : Find a policy that has disjoint final classes of states.*

With Question 7 you know that the optimal average-reward g^* does not depend on the initial state s_1.

Question 9 : Show that there exist some quantities $h(s)$ such that g^* and the $h(s)$ satisfy together

$$\forall s, \quad g^* + h(s) = \max_a \sum_{s'} P(s'|s,a) \left[R(s,a,s') + h(s') \right]$$

Question 10 : Rewrite the previous equation in a vectorial form, with $h \in \mathbb{R}^{|\mathcal{S}|}$, $R_\pi = \big(R(s,\pi(s),s')\big)_{(s,s')}$, and $P_\pi = \big(P(s'|s,\pi(s))\big)_{(s,s')}$ for some policy π.

4 Computational part

Now you have to write a program in Python to implement an algorithm that computes the optimal policy in the average-reward setting. Your work provides the following functions :

- `optimal_gain_gld(N, K, W, L, CR, CT)` computes and returns the optimal average-reward g^* of the MDP with the corresponding parameters. The choice of the algorithm (value iteration, policy iteration, ...) is yours.

- Bonus : `play_gld(N, K, W, L, CR, CT)` enables to play the game interactively. The current state and the rewards are printed on the screen and the user can choose the action at each time.

Parameters reminder :

- N : number of cells.
- K : number of cells the dinosaur eats when activated.
- W : amont of points won when the dinosaur lays an egg.
- L : amont of points lost when the dinosaur does not eat anything when activated.
- CR : amont of points lost when choosing action (BR).
- CT : amont of points lost when choosing action (BT).